Dell Storage Center Command Set 7.1 for Windows PowerShell

Administrator’s Guide

Regulatory Model: 680-037-006
September 2021
Rev. B

Notes, cautions, and warnings

ⓘ **NOTE:** A NOTE indicates important information that helps you make better use of your product.

⚠ **CAUTION: A CAUTION indicates either potential damage to hardware or loss of data and tells you how to avoid the problem.**

⚠ **WARNING: A WARNING indicates a potential for property damage, personal injury, or death.**

Preface

This guide provides instructions for installing and using the Dell Storage Center Command Set for Windows PowerShell.

Revision History

Table 1. Document Revision History

Revision	Date	Description
A	September 2014	Initial release
B	September 2021	Updated Windows PowerShell requirements

Audience

Target audiences for this document are Dell storage administrators with knowledge of Windows PowerShell.

Related Publications

The following documents contain relevant information when using the Dell Storage Center Command Set:

- *Storage Center System Manager Administrator's Guide*

  Describes the Storage Center System Manager software that manages an individual Storage Center.

- *Dell Storage Center Command Utility Reference Guide*

  Provides instructions for using the Dell Storage Center Command Utility. The Command Utility provides a command-line interface (CLI) to enable management of Dell Storage Center functionality on Windows, Linux, Solaris, and AIX platforms.

- *Dell Storage Center Software Update Guide*

  Describes how to update Dell Storage Center software from an earlier version to the current version.

- *Dell Storage Center Release Notes*

  Contains information about features and open and resolved issues for a particular product version.

Contents

1

Getting Started

This section provides a brief overview of the Dell Storage Center Command Set snapin, lists requirements for using the command set, and briefly describes features and enhancements that are new with the release.

Topics:

- Introduction
- Requirements
- Installing the Command Set

Introduction

Dell Storage Center Command Set provides access to cmdlets and scripting objects that interact with a Dell Storage Center via Windows PowerShell interactive shell, scripts, and PowerShell hosting applications.

For detailed information about all available cmdlets, access the cmdlet online help. For instructions, see Viewing and Printing Command Line Help.

Requirements

Dell Storage Center Command Set requires the following components.

Requirement	Description
Storage Center	Version 5.5.6 or later.
.NET Framework	Version 3.5 or later.
Windows PowerShell	Version 2.0 to 5.1. ⓘ **NOTE:** Windows PowerShell versions 6.0 and later might work with Dell Storage Center Command Set 7.1, but they have not been tested for compatibility. ⓘ **NOTE:** Install Windows PowerShell before installing the Dell Storage Center Command Set snapin.
Storage Center User Privileges	Cmdlet access to Storage Center folders, volumes, and views depends on the Storage Center user privilege and the group membership: ● **Administrators** have full system control. ● **Volume Managers** have access folders associated with their assigned User Groups. They can create volumes in the allowed volume folders and map them to existing servers in the allowed server folders. ● **Reporters** have read-only access to folders associated with their assigned user group.
Script Privileges	Script file execution requires the following privileges: ● Write access to either the directory in which the PowerShell executable is located or the directory in which the script file is located. ● Appropriate Storage Center access privileges for the user running the script to allow all cmdlets to execute without access errors. If the user running a script lacks the required privileges for the script, the script will fail. ⓘ **NOTE:** Windows PowerShell connections to Disk Manager and other Microsoft products are provided by Microsoft to enable full disk control between a Storage Center and a Windows server.

Installing the Command Set

Complete the following steps to install the Dell Storage Center Command Set.

1. Navigate to the location of the Dell Storage Center Command Set setup file.
2. Double-click the setup file. The installation wizard starts.
3. To complete installation, follow the installation wizard instructions.

2

Using the Command Set

This section provides instructions for installing and setting up Dell Storage Center Command Set for Windows PowerShell and provides information for getting started with the Command Set.

Topics:

Accessing the Dell Storage Center Command Set

You can access the Dell Storage Center Command Set using the following methods:

- Shell access
- Console access

Using the Dell Storage Center Command Set Shell

To access the Dell Storage Center Command Set using the Command Set shell, select:

Start > **All Programs** > **Dell Compellent** > **Storage Center PowerShell Snapin** > **Dell Compellent Storage Center Command Set Shell**

The shell provides a shortcut to a Windows PowerShell console with the Dell Storage Center Command Set already loaded. Dell recommends using this method to access the Dell Storage Center Command Set.

Using the Windows PowerShell Console

To access the Dell Storage Center Command Set using the standard Windows PowerShell console:

1. Access the Windows PowerShell console.
2. Open the console and load the Dell Storage Center Command Set snapin by typing the following command:
   ```
   Add-PSSnapin Compellent.StorageCenter.PSSnapin
   ```

The snapin must be loaded each time the Windows PowerShell console is opened prior to using the Dell Storage Center Command Set.

NOTE: See Migrating Scripts from 5.x for examples of loading two versions of the Command Set to facilitate migrating from version 5.x to version 6.x or 7.x.

Connecting to a Storage Center

Storage Center cmdlets require a connection to a Storage Center. This section details how to establish a connection to a Storage Center. In addition, this section describes how to use SecureString input to safeguard password information.

Establishing a Connection

The **Get-SCConnection** cmdlet establishes a connection to a Storage Center. You can provide connection information to the cmdlet in the following ways:

- Using a variable
- Using a ConnectionName object
- Interactively

Save Connection Information in a Variable

Connection information can be stored in a variable and then used on subsequent cmdlets. For example:

```
$connection = Get-SCConnection -HostName mmt1 -User Admin -Password $pass
```

The *$connection* variable is used with the `-Connection` parameter to connect to a Storage Center. For example:

```
Get-SCVolume -Connection $connection
```

Save Connection Information to a -ConnectionName Object

Connection information can be saved to a `-ConnectionName` object using the `-Save` parameter with the `Get-SCConnection` command. For example:

```
Get-SCConnection -HostName mmt1 -User Admin -Password $pass -Save connectionname
```

The saved object is used with the `-ConnectionName` parameter to connect to a Storage Center. For example:

```
Get-SCVolume -ConnectionName connectionname
```

Enter Connection Information Interactively

If you do not provide all required parameters (`-HostName`, `-User`, `-Password`) for the **Get-SCConnection** command, the system prompts for the missing information. This method still requires storing connection information in a variable or as a connection name.

Creating Default Connections

To create a default Storage Center connection, use the `-Default` or `-DestinationDefault` parameters when executing the `Get-SCConnection` cmdlet.

- Saving a connection as the default allows Storage Center cmdlets to be executed without specifying the `-Connection` or `-ConnectionName` parameters.
- Saving a destination default connection allows Storage Center cmdlets that require a destination connection to be executed without specifying the `-DestConnection` or `-DestConnectionName` parameters.

For details and examples, see Viewing and Printing Command Line Help.

Creating SecureString Passwords

SecureString objects are used in Windows PowerShell cmdlets as a security measure. SecureStrings are encrypted in memory and the content of a SecureString variable cannot be displayed. For this reason, SecureString objects are commonly used for sensitive information such as passwords.

> **NOTE:** If security is not a prime concern, passwords can be entered directly on the command line without obscuring the text. For example:
>
> ```
> $pass = ConvertTo-SecureString "mypassword" -AsPlainText -Force
> ```

SecureString Input

Create a variable with SecureString content by entering the following command in the PowerShell console:

```
$pass = Read-Host -AsSecureString -Prompt "Please provide password"
```

The PowerShell console returns the following prompt:

```
Please provide password:
```

The password is obscured by asterisks (`*`) and stored in the variable.

Stored Values

SecureString values can be stored for later access. The encryption used for the SecureString is based on the logged-on user credentials. The credentials help prevent another user from accessing the stored value and deriving the password. However, the saved data should be kept in a secure location and NTFS permissions might also be used to further protect it from unwanted access.

To store your password in an encrypted file:

1. Enter the following command:

   ```
   $pass = Read-Host -AsSecureString -Prompt "Please provide password"
   ```

   PowerShell prompts for a password.

   ```
   Please provide password:
   ```

2. Enter your password.
3. Save your password to a text file:

   ```
   ConvertFrom-SecureString $pass > C:\password.txt
   ```

A script can now access this file using the following command:

```
$pass = ConvertTo-SecureString (Get-Content C:\password.txt)
```

Using PowerShell Objects

Windows PowerShell cmdlet parameters, including those available with the Dell Storage Center Command Set, accept objects as arguments. Objects include standard objects such as string and integer objects, as well as Storage Center specific objects.

Store Objects in Variables

Objects can be created and stored in variables. For example, an SCVolume object can be created and stored in a variable using the `Get-SCVolume` command:

```
$myvolume = Get-SCVolume -Name “My Volume”
```

The SCVolume object can be moved using the following commands:

```
$myvolume = Get-SCVolume -Name “My Volume”
$newfolder = Get-SCVolumeFolder -Name “My VolumeFolder”
Set-SCVolume -SCVolume $myvolume -ParentSCVolumeFolder $newfolder
```

Access Object Information

Use the `Get-Member` command to access object information. The `Get-Member` command accepts pipeline input from commands that return objects. For example, the `Get-SCVolume` command returns an SCVolume object. To view SCVolume object information, enter the following command:

```
Get-SCVolume|Get-Member
```

This command returns a list of methods and properties for the SCVolume object.

Object Descriptions

The following table lists and describes common PowerShell objects as well as Storage Center specific objects.

Table 2. PowerShell and Storage Center Specific Objects

Object	Description
Boolean	Represents a true or false value. For true, type 1 or $true. For false, type 0 or $false.
DataVolume	Represents a logical volume in Windows.
DateTime	Represents dates and times with values ranging from 12:00:00 (midnight), January 1, 0001 AD (CE) through 11:59:59 p.m. December 31, 9999 AD (CE). Time values are measured in 100-nanosecond units called ticks.
DiskDeviceObject	Represents a disk drive in Windows.
Int32	Represents a 32-bit signed integer. Signed integers can be either positive or negative.
Int64	Represents a 64-bit signed integer. Signed integers can be either positive or negative.
SCAlert	Represents a Storage Center alert.
SCAsyncReplication	Represents a Storage Center asynchronous replication.
SCCmm	Represents a Storage Center copy/mirror/migrate operation.
SCConnection	Represents Storage Center connection information.
SCController	Represents a Storage Center controller.
SCDisk	Represents a Storage Center disk.
SCDiskFolder	Represents a Storage Center disk folder.

Table 2. PowerShell and Storage Center Specific Objects (continued)

Object	Description
SCLiveVolume	Represents a Storage Center Live Volume.
SCOSType	Represents a Storage Center operating system type.
SCPort	Represents a Storage Center controller port.
SCReplay	Represents a Storage Center Replay.
SCReplayProfile	Represents a Storage Center Replay profile.
SCReplaySchedule	Represents a Storage Center Replay schedule.
SCReplicationQos	Represents a Storage Center replication quality of service (QoS) definition.
SCServer	Represents a Storage Center server.
SCServerFolder	Represents a Storage Center server folder.
SCStorageCenter	Represents a Storage Center.
SCStorageProfile	Represents a Storage Center storage profile.
SCStorageType	Represents a Storage Center storage type.
SCUser	Represents a Storage Center user account.
SCVolume	Represents a Storage Center volume.
SCVolumeFolder	Represents a Storage Center volume folder.
SCVolumeMap	Represents a Storage Center volume-to-server mapping.
SecureString	Represents an encrypted string for private data.
String	Represents a combination of alphanumeric characters. Strings that contain the space character must be enclosed in quotation marks.
UInt32	Represents a 32-bit unsigned integer. Unsigned integers cannot be negative.
UInt64	Represents a 64-bit unsigned integer. Unsigned integers cannot be negative.

Available Commands

The following table lists and describes available Dell Storage Center Command Set cmdlets.

Table 3. Command Set cmdlets

Command	Description
Acknowledge-SCAlert	Marks a Storage Center alert as acknowledged.
Add-CMLVolumeAccessPath	Same as Add-VolumeAccessPath. Avoids name collisions.
Add-SCReplayProfileRule	Adds a new SCReplaySchedule to a Replay profile. The SCReplaySchedule can be created using the `New-SCReplaySchedule` cmdlet or copied from an existing SCReplaySchedule retrieved with the `Get-SCReplaySchedule` cmdlet.
Add-SCServerPort	Adds additional world-wide name (WWN) or iSCSI addresses to an existing server definition.
Add-VolumeAccessPath	Adds a drive letter or mount point to an existing volume.
CancelSwapRole-SCLiveVolume	Cancels the swap role of a Live Volume.
ConvertTo-SCAsyncReplication	Converts a Live Volume to an asynchronous replication.
ConvertTo-SCLiveVolume	Converts an asynchronous replication to a Live Volume.

Table 3. Command Set cmdlets (continued)

Command	Description
Expand-SCVolume	Expands a Storage Center by additional reported capacity.
Get-CMLDiskDevice	Same as Get-DiskDevice. Avoids name collisions.
Get-CMLNextFreeDriveLetter	Same as Get-NextFreeDriveLetter. Avoids name collisions.
Get-CMLVolume	Same as Get-Volume. Avoids name collisions.
Get-DiskDevice	Retrieves disk drive information from the Windows Virtual Disk Service.
Get-NextFreeDriveLetter	Queries the Windows Virtual Disk Service for the next available drive letter that could be used to assign as a volume path.
Get-SCAlert	Retrieves all alerts that were generated on the controller with a specified serial number.
Get-SCAsyncReplication	Returns asynchronous replications, including those that support Live Volume replications. By default, the `Get-SCAsyncReplication` cmdlet does not return this type of replication. The LiveVolume parameter includes asynchronous replications supporting Live Volume replications to preserve previous behavior.
Get-SCCmm	Retrieves Copy, Mirror, or Migrate operations from Storage Center.
Get-SCCommandSetVersion	Returns a system version object containing the version number of the currently loaded Dell Storage Center Command Set.
Get-SCConnection	Establishes connection credentials to a Storage Center system.
Get-SCDisk	Retrieves disk information from Storage Center.
Get-SCDiskFolder	Retrieves disk folder information from Storage Center.
Get-SCLiveVolume	Retrieves Live Volume operations from Storage Center (`SourceVolumeName` parameter).
Get-SCOSType	Retrieves the defined server operating system type.
Get-SCPort	Retrieves all front-end primary port information from Storage Center.
Get-SCRecycleBinVolume	Retrieves the volumes in the Storage Center recycle bin (TotalDiskSpace attribute).
Get-SCReplay	Retrieves Replay information from Storage Center.
Get-SCReplayProfile	Retrieves all available Replay profiles from Storage Center.
Get-SCReplaySchedule	Retrieves a Replay schedule by index or all the Replay schedules for a Replay profile.
Get-SCReplicationsQos	Retrieves Replication quality of service (QoS) definitions from Storage Center.
Get-SCServer	Retrieves server definitions from Storage Center.
Get-SCServerFolder	Retrieves server folder definitions from Storage Center.
Get-SCStorageCenter	Retrieves Storage Center information.
Get-SCStorageProfile	Retrieves all available Storage Profiles from Storage Center.
Get-SCStorageType	Retrieves available storage types from Storage Center.
Get-SCUser	Allows administrator-level users to retrieve all users from Storage Center.
Get-SCVolume	Retrieves volumes from Storage Center. Includes a TotalDiskSpace attribute, which represents the Volume Space Consumed plus the Replay overhead. ⓘ **NOTE:** In the case of a view volume, this number includes the space borrowed from the underlying Replay.

Table 3. Command Set cmdlets (continued)

Command	Description
Get-SCVolumeFolder	Retrieves volume folders from Storage Center.
Get-SCVolumeMap	Retrieves volume maps from Storage Center.
New-CMLVolume	Same as New-Volume. Avoids name collisions.
New-SCAsyncReplication	Creates a new asynchronous replication (`SCDestinationVolume` parameter).
New-SCCmmCopy	Creates a Copy, Mirror, or Migrate operation on Storage Center to copy all data from one volume to another.
New-SCCmmMigrate	Creates a Copy, Mirror, or Migrate operation on Storage Center to migrate all data from one volume to another.
New-SCCmmMirror	Creates a Copy, Mirror, or Migrate operation on Storage Center to mirror all data from one volume to another.
New-SCLiveVolume	Creates a new Live Volume.
New-SCReplay	● Creates a new Replay on a volume (`SCReplayProfile` parameter). ● The expire time and freeze time returned from the `Get-SCReplay` cmdlet formats the date values using the Windows regional settings. Expire Time displays `Never Expire` if the Replay is set to never expire.
New-SCReplayProfile	Creates a new Replay profile. Adds a single rule when a profile is created using the optional `Rule` parameter. To add multiple rules to the profile, see the `Add-SCReplayProfileRule` cmdlet. To remove rules from a Replay profile, see the `Remove-SCReplayProfile` cmdlet.
New-SCReplaySchedule	Creates a new SCReplaySchedule instance. The result of this cmdlet can be passed as the `-SCReplaySchedule` parameter in the `New-SCReplayProfile` and the `Add-SCReplayProfileRule` cmdlets. ⓘ **NOTE:** The Replay schedule is not created on the Storage Center until it is associated with a Replay profile.
New-SCReplicationQos	Creates a new quality of service (QoS) definition.
New-SCServer	Defines a new server.
New-SCServerFolder	Creates a new volume folder that can contain other volumes and folders.
New-SCStorageProfile	Creates a new Storage Profile.
New-SCUser	Creates a new user account on Storage Center.
New-SCVolume	Creates a new volume on Storage Center. Does not display a warning when the volume name conflicts with a volume in the recycle bin.
New-SCVolumeFolder	Creates a new volume folder.
New-SCVolumeMap	Exposes volumes to servers by creating a path and LUN assignment for a volume between a controller port and server port.
New-Volume	Communicates with the Windows Virtual Disk Manager service to create a new volume on a disk.
Pause-SCAsyncReplication	Pauses the asynchronous replication.
Pause-SCCmm	Pauses the Copy, Mirror, or Migrate operation.
Preallocate-SCVolume	Preallocates space for a volume. Preallocating storage physically assigns storage to a volume before use by the server.
Remove-CMLVolumeAcccessPath	Same as Remove-VolumeAcccessPath. Avoids name collisions.
Remove-SCAsyncReplication	Stops a replication operation on Storage Center.

Table 3. Command Set cmdlets (continued)

Command	Description
Remove-SCCmm	Removes a Copy, Mirror, or Migrate operation from Storage Center by stopping the operation.
Remove-SCConnection	Removes Storage Center connection settings saved using the `-Save` parameter from the `Get-SCConnection` cmdlet.
Remove-SCLiveVolume	Removes a Live Volume from Storage Center.
Remove-SCRecycleBinVolume	Removes (permanently deletes) one or more volumes from the Storage Center recycle bin.
Remove-SCReplay	Sets Replays on Storage Center to expire immediately.
Remove-SCReplayProfile	Removes the Replay profile from the Storage Center. Displays an error if the profile is currently applied to any volumes. The `-Force` parameter removes the Replay profile from assigned volumes and then removes the Replay profile. Attempting to remove a system-created Replay profile returns an error.
Remove-SCReplaySchedule	Removes a Replay schedule from a Replay profile.
Remove-SCReplicationQos	Deletes a quality of service (QoS) definition.
Remove-SCServer	Removes server object definitions from Storage Center.
Remove-SCServerFolder	Deletes an empty server folder.
Remove-SCServerPort	Removes a Fibre Channel or iSCSI port from a server definition.
Remove-SCStorageProfile	Removes the Storage Profile from the Storage Center. Displays an error if the Storage Profile is currently applied to any volumes. Attempting to remove a system-created Storage Profile returns an error.
Remove-SCUser	Allows administrator-level users to remove user accounts from Storage Center.
Remove-SCVolume	Deletes a volume, removing volume mappings if necessary.
Remove-SCVolumeFolder	Deletes an empty volume folder.
Remove-SCVolumeMap	Deletes maps between a volume and a server.
Remove-SCVolumeReplayProfile	Removes a Replay profile from a volume.
Remove-VolumeAccessPath	Communicates with the Microsoft Virtual Disk Service to remove a volume drive letter or mount point from a volume.
Rescan-CMLDiskDevice	Same as Rescan-DiskDevice. Avoids name collisions.
Rescan-DiskDevice	Communicates with the Microsoft Virtual Disk Service to cause a rescan for changes to the disk devices attached to the server.
Restore-SCRecycleBinVolume	Restores one or more volumes from the recycle bin into its parent folder.
Resume-SCAsyncReplication	Resumes the asynchronous replication.
Resume-SCCmm	Resumes the Copy, Mirror, or Migrate operation.
Set-CMLDiskDevice	Same as Set-DiskDevice. Avoids name collisions.
Set-CMLVolume	Same as Set-Volume. Avoids name collisions.
Set-DiskDevice	Sets the online status of a disk device.
Set-SCVolumeStorageProfile	Applies a Storage Profile to a volume.
Set-SCAsyncReplication	Sets attributes of an asynchronous replication operation on Storage Center (`Name` parameter).
Set-SCCmm	Sets the priority of a Copy, Mirror, or Migrate operation.

Table 3. Command Set cmdlets (continued)

Command	Description
Set-SCDisk	Sets attributes of a disk object on Storage Center.
Set-SCLiveVolume	Sets attributes of a Live Volume operation on Storage Center (`Name` and `ReplicateToLowestTier` parameters).
Set-SCReplay	Sets the description and expiration time for an existing Replay. The Expiration Time and Never Expires switches cannot be specified at the same time.
Set-SCReplayProfile	Sets the attributes of a Replay profile.
Set-SCReplicationQos	Sets attributes of a quality of service (QoS) definition.
Set-SCServer	Set attributes of a server definition such as name and notes.
Set-SCServerFolder	Set attributes of a server folder object such as name and parent folder.
Set-SCStorageProfile	Sets the attributes of a Storage Profile.
Set-SCUser	Sets attributes of user accounts on a Storage Center (`Password` parameter).
Set-SCVolume	Sets a new name for a volume.
Set-SCVolumeFolder	Sets attributes of a volume folder such as name and parent folder.
Set-SCVolumeReplayProfile	Applies a Replay profile to a volume.
Set-Volume	Sets or clears flags of a volume on a Windows server.
Swap Role-SCLiveVolume	Swaps role of a Live Volume.

For more information, see Viewing and Printing Command Line Help.

Entering Single Commands

Windows PowerShell provides an interactive command prompt that can be used to execute single commands.

Open the Dell Storage Center Command Set shell and enter a single command. For example:

```
Get-SCConnection -HostName mmt1 -User Admin -Password $pass
```

Creating Scripts

A script is a text file that contains one or more Windows PowerShell commands. In PowerShell version 1.0 scripting is most often done in Notepad. PowerShell version 2.0 includes the Integrated Scripting Environment (ISE), which provides a graphical user interface (GUI) for script editing. The GUI includes advanced features such as syntax color coding and autocompletion of cmdlet names.

Follow these guidelines for creating script files:

- Include one command per line.
- Use the pound sign (#) as the first character on a line to denote a comment.

During installation, sample scripts are installed in the `\Script Samples` location of the installation directory. For example:

```
C:\Program Files (x86)\Dell\Storage Center PowerShell Command Set\Script Samples
```

These sample scripts should be used as templates that are edited to conform to specific requirements. Sample scripts are a starting point for script development and are not guaranteed to work in all situations. Installed scripts offer samples for:

- Creating new volumes and an SQL server database
- Creating new volumes and an Exchange mailbox storage group
- Creating new user accounts for everyone in a user group

- Exposing a Replay to a backup server for offloading a backup to tape

Running Scripts

For script file execution, you must have all privileges required by commands in the script or the script will fail. Follow these guidelines for running a script.

- By default, Windows PowerShell script execution policy is set to restricted. For information about enabling scripts, enter either:
  - `Get-Help Set-ExecutionPolicy`
  - `Get-Help about_signing`
- Run scripts from the Dell Storage Center Command Set Shell or from the PowerShell console with the Dell Storage Center Command Set snapin loaded.
- The Dell Storage Center Command Set does not support running multiple scripts concurrently.
- Specify the entire file path of your script:

  ```
  C:\scripts\myscript.ps1
  ```

- If the path to your script includes blank spaces, type an ampersand (`&`) followed by the path enclosed in quotation marks:

  ```
  &"C:\my scripts\myscript.ps1"
  ```

- To run a script from the current directory, use `.\`:

  ```
  .\myscript.ps1
  ```

- Include the folder containing your script in your Windows path, then type the name of the script:

  ```
  myscript.ps1
  ```

Storage Center Snapin Utility

The Storage Center Snapin Utility module provides a library of common functions. The utility module was developed using PowerShell scripting.

NOTE: The Storage Center PowerShell Snapin must be added to the current PowerShell session to use the functions provided by the utility.

Table 4. Snapin Utility Commands

Command	Description
Get-CMLDiskSignature	Retrieves the disk signatures of all disks or the disk specified in the index parameter.
Get-CMLDuplicateDiskSignature	Returns groups of disks with identical signatures. This duplication can occur when a view volume for a clustered shared volume (CSV) is mounted to a server. Windows will not update the signature for a CSV.
Set-CMLDiskSignature	Sets the disk signature of the disk specified by the index parameter.
Set-CMLVolumeSignature	Sets the disk signature for a specified volume by mapping the volume on the computer identified by the `ComputerName` parameter, changing the signature and removing the mapping. This feature is particularly useful for automating the process of fixing the disk collision issue when attempting to mount the view volume of a CSV on a cluster node. To avoid a collision, the `ComputerName` parameter must refer to a computer that is not a node in the cluster. When this function completes successfully, the view volume can then be mapped to a node in the cluster without a disk signature collision.

Formatting and Exporting Command Output

Windows PowerShell provides commands that can either format the way information appears in the console or export information to an xml, csv, or text file. These commands accept the output of other commands as input from the pipeline.

Format Command Output in the Console

By default, Dell Storage Center Command Set cmdlets print information to the console in a table format. The `Format-List` command formats information as a list. For example, the following command displays a list of all properties for a Storage Center volume named `MyVolume`:

```
Get-SCVolume -Name MyVolume|Format-List
```

For more information about displaying information on the console, type the following command:

```
Get-Help about_display
```

Export Command Output to a File

Command output can be stored in standard file formats. The following examples show how to save the information of a Storage Center volume named `MyVolume` as an `xml`, `csv`, or `txt` file.

To save information in an `xml` file:

```
Get-SCVolume -Name MyVolume|Export-Clixml -path “C:\volumeInformation\MyVolume.xml”
```

To save information to a `csv` file:

```
Get-SCVolume -Name MyVolume|Export-Csv -path “C:\volumeInformation\MyVolume.csv”
```

To save information as a `txt` file:

```
Get-SCVolume -Name MyVolume|Format-List|Out-File -filePath “C:
\volumeInformation\MyVolume.txt”
```

Viewing and Printing Command-Line Help

The Dell Storage Center Command Set provides help for the commands. You can access this help from the command line.

Access help information for each cmdlet in the Command Set Shell

Enter the `Get-Help` command. For example:

```
Get-Help Get-SCVolume
```

This command returns the following information:

- Name
- Synopsis
- Syntax
- Detailed description
- Related Links
- Remarks

View additional information

Enter the `-detailed` option. For example:

```
Get-Help Get-SCVolume -detailed
```

This command returns all the information listed previously, plus additional information about parameters and examples.

View all information available for a cmdlet

Enter the `-full` option. For example:

```
Get-Help Get-SCVolume -full
```

View a list of all available Storage Center Snapin cmdlets

Enter the `Get` command. For example:

```
Get-Command -Module Compellent.StorageCenter.PSSnapin
```

Print help for all cmdlets

Enter the `-full` option and export output to a file. For example:

```
Get-Command -Module Compellent.StorageCenter.PSSnapin | %{Get-Help $_.Name -Full}
```

A

Migrating Scripts from Version 5.x

This appendix provides information about migrating from Dell Storage Center Command Set version 5.x to 7.x. Scripts written with version 6.x are compatible with version 7.

Topics:

- Syntax Modification Examples
- Migrating Existing Scripts
- Snapins for Versions 5.x and 7.x Script Examples

Syntax Modification Examples

At release 6.x, cmdlet syntax was modified to make parameters consistent, increase the ability to pipeline results, and simplify cmdlet syntax. No additional syntax modifications were made in version 7.x. The following examples illustrate how modifications achieve these goals.

Example One: Expand-SCVolume

`Expand-SCVolume` illustrates differences in cmdlet syntax between versions 5.x and 6.x/7.x. The version 6.x/7.x syntax is cleaner and easier to use than the version 5.x syntax.

Expand-SCVolume Syntax Version 5.x

```
Expand-SCVolume [[-NewSize] <String>] [[-ExpandBy] <String>] [[-Name] <String>] [-
Index <UInt32>] [-ParentFolder <String>] [-LogicalPath <String>] [-Notes <String>]
[-Size <String>] [ReadCacheEnabled [<Boolean>]] [-ReadAheadCacheEnabled [<Boolean>]]
[-WriteCacheEnabled [<Boolean>]] [-ActiveController <UInt32>] [-Status <String>]
[-SerialNumber <String>] [-StorageType <String>] [-IsReplicated [<Boolean>]] [-
IsCopying [<Boolean>]] [-IsMirrored [<Boolean>]] [-SCStorageProfile <SCStorageProfile>]
[-StorageProfileIndex <UInt32>] [-StorageProfileName <String>] [-ReplayProfileIndex
<UInt32>] [-ReplayProfileName <String>] [-SCReplayProfile <SCReplayProfile>] [-SCVolume
<SCVolume>] [-Connection <SCConnection>] [-ConnectionName <String>] [-WarningAction
<ActionPreference>] [-WarningVariable <String>] [-WhatIf] [-Confirm] [<CommonParameters>]
```

Expand-SCVolume Syntax Version 6.x or 7.x

```
Expand-SCVolume [-SCVolume] <SCVolume> [[-NewSize] <String>] [[-ExpandBy] <String>]
[-Connection <SCConnection>] [-ConnectionName <String>] [-WhatIf] [-Confirm]
[<CommonParameters>]
```

The parameters that were removed had been used to select any volumes to be expanded. These parameters are referred to as filtering parameters, and are available in the `Get-SCVolume` cmdlet.

The following examples show how filtering is performed by the `Get-SCVolume` cmdlet in version 6.x.

- Example of using `Expand-SCVolume` version 5.x:

  ```
  Expand-SCVolume -Name TestVolume -ExpandBy 10G
  ```

- Example of using `Expand-SCVolume` version 6.x with PowerShell variables:

  ```
  $vol = Get-SCVolume -Name TestVolume Expand-SCVolume -SCVolume $vol -ExpandBy 10G
  ```

- Example of using `Expand-SCVolume` version 6.x with embedded cmdlets:

  ```
  Expand-SCVolume -SCVolume (Get-SCVolume -Name TestVolume) -ExpandBy 10G
  ```

- Example of using `Expand-SCVolume` version 6.x with cmdlets that have pipelines:

  ```
  Get-SCVolume -Name TestVolume | Expand-SCVolume -ExpandBy 10G
  ```

The last three examples also demonstrate the different ways in which the result of the `Get-SCVolume` cmdlet can be used as input to the `Expand-SCVolume` cmdlet.

In a similar way, other cmdlets have been modified to remove filtering and provide pipelining.

Example Two: Set-SCVolume

`Set-SCVolume` illustrates differences in cmdlet syntax between versions 5.x and 6.x/7.x. The version 6.x/7.x syntax is cleaner and easier to use than the version 5.x syntax.

Set-SCVolume Syntax Version 5.x

```
Set-SCVolume [-NewName <String>] [-NewNotes <String>] [-EnableReadCache [<Boolean>]]
[-EnableReadAheadCache [<Boolean>]] [-EnableWriteCache [<Boolean>]] [-NewParentFolder
<String>] [-NewParentSCVolumeFolder <SCVolumeFolder>] [[-Name] <String>] [-Index
<UInt32>] [-ParentFolder <String>] [-LogicalPath <String>] [-Notes <String>] [-
Size <String>] [-ReadCacheEnabled [<Boolean>]] [-ReadAheadCacheEnabled [<Boolean>]]
[-WriteCacheEnabled [<Boolean>]] [-ActiveController <UInt32>] [-Status <String>]
[-SerialNumber <String>] [-StorageType <String>] [-IsReplicated [<Boolean>]] [-
IsCopying [<Boolean>]] [-IsMirrored [<Boolean>]] [-SCStorageProfile <SCStorageProfile>]
StorageProfileIndex <UInt32>] [-StorageProfileName <String>] [-ReplayProfileIndex
<UInt32>] [-ReplayProfileName <String>] [-SCReplayProfile <SCReplayProfile>] [-SCVolume
<SCVolume>] [-Connection <SCConnection>] [-ConnectionName <String>] [-WarningAction
<ActionPreference>] [-WarningVariable <String>] [-WhatIf] [-Confirm] [<CommonParameters>]
```

Set-SCVolume Syntax Version 6.x/7.x

```
Set-SCVolume [-SCVolume] <SCVolume> [-Name <String>] [-Notes <String>] [-EnableReadCache
[<Boolean>]]
[-EnableReadAheadCache [<Boolean>]] [-EnableWriteCache [<Boolean>]] [-
ParentSCVolumeFolder <SCVolumeFolder>]
[-Connection <SCConnection>] [-ConnectionName <String>] [-WhatIf] [-Confirm]
[<CommonParameters>]
```

In this case, the filtering commands were removed and replaced by the `-SCVolume` parameter.

Migrating Existing Scripts

Syntax changes between the versions impact existing scripts. Scripts written for version 5.x will not run using version 6.x or 7.x without modification. Dell recommends modifying old scripts to run with version 7.x as soon as possible.

As an interim solution, the version 6.x and 7.x snapins are designed to coexist with the version 5.x snapin. To work together, the changes described in the following sections were made in versions 6.x and 7.x.

Install Procedure Modification

To preserve version 5.x availability, the version 6.x and 7.x install procedures do not automatically remove version 5.x. When version 5.x is no longer needed, you must explicitly remove the version 5.x software by going to **Windows Control Panel** > **Programs and Features** and manually uninstalling the older version.

New Name for PowerShell Snapin

The snapin name for version 6.x/7.x is **Compellent.StorageCenter.PSSnapin**. For version 5.x, the snapin name remains **Compellent.StorageCenter.Scripting**.

- Scripts using the version 5.x snapin should include the **Add-PSSnapin Compellent.StorageCenter.Scripting** cmdlet.
- Scripts using the version 6.x or 7.x snapin should include the **Add-PSSnapin Compellent.StorageCenter.PSSnapin** cmdlet.

For script examples, see Snapins for Versions 5.x and 7.x Script Examples.

Internal Object Type Modifications

Names and definitions for internal types have changed. For example, in version 5.x the name of the Volume type was **SCCommandSet.SCObjects.Volume**; in version 6.x/7.x the name of the type is **SCCommandSet.SCObjects.SCVolume**.

These changes mean that you cannot pass objects between versions of commands. For example, you cannot take the result of the version 5.x `Get-SCVolume` cmdlet and pipe it into the version 6.x/7.x `Set-SCVolume` cmdlet.

Snapins for Versions 5.x and 7.x Script Examples

Dell Storage Center Command Set 7.x allows you to have snapins for versions 5.x and 7.x loaded at the same time to accommodate cmdlet modifications. The following script examples demonstrate using the `Add-PSSnapin` cmdlet when migrating from version 5.x to 7.x.

- The script examples are written to run in a PowerShell console; you must explicitly load the Storage Center snapins. If scripts are run from a Dell Storage Center Command Set Shell, one of the snapins will already be loaded (depending on the version of the Command Set Shell run) and the script examples will have to be modified appropriately.
- If a script needs only one version of the snapin, only that snapin should be loaded.
- If a script needs both versions of the snapin, they can be loaded one at a time or at the same time.

NOTE: If both snapins are loaded at the same time, cmdlets duplicated in both versions should be fully qualified with the snapin name.

Example Using Version 5.x Snapin Only

```
Add-PSSnapin Compellent.StorageCenter.Scripting
Expand-SCVolume -Name TestVolume -ExpandBy 10G
Remove-PSSnapin Compellent.StorageCenter.Scripting
```

Example Using Version 7.x Snapin Only

```
Add-PSSnapin Compellent.StorageCenter.PSSnapin
Get-SCVolume -Name TestVolume | Expand-SCVolume -ExpandBy 10G
Remove-PSSnapin Compellent.StorageCenter.PSSnapin
```

Example with One Snapin Loaded

```
Add-PSSnapin Compellent.StorageCenter.Scripting
Expand-SCVolume -Name TestVolume -ExpandBy 10G
Remove-PSSnapin Compellent.StorageCenter.Scripting
Add-PSSnapin Compellent.StorageCenter.PSSnapin
Get-SCVolume -Name TestVolume | Expand-SCVolume -ExpandBy 10G
Remove-PSSnapin Compellent.StorageCenter.PSSnapin
```

Example with Both Snapins Loaded

```
Add-PSSnapin Compellent.StorageCenter.Scripting
Add-PSSnapin Compellent.StorageCenter.PSSnapin
Compellent.StorageCenter.Scripting\Expand-SCVolume -Name TestVolume -ExpandBy 10G
Compellent.StorageCenter.PSSnapin\Get-SCVolume -Name TestVolume |
Compellent.StorageCenter.PSSnapin\Expand-SCVolume
-ExpandBy 10G
Remove-PSSnapin Compellent.StorageCenter.PSSnapin
Remove-PSSnapin Compellent.StorageCenter.Scripting
```