



NetApp Element

PowerShell Tools User Guide

Version 1.6

July 2020 | 215-14886_A0
doccomments@netapp.com

Contents

Introduction.....3

How to use NetApp Element PowerShell Tools.....4

 Listing available functions.....4

 Accessing embedded help.....5

 Parameter sets.....5

 Managing connections to an Element storage cluster.....5

 Connecting to an Element storage cluster.....6

 Connecting to an Element storage node.....6

 Disconnecting from an Element storage cluster or node.....7

 Changing API versions.....7

 Global variables for all functions.....8

 Common parameters.....9

 Return object descriptions.....9

 Accessing return value reference documentation.....9

 Accessing return values using Get-Help.....9

 Leveraging Get-Member to inspect return objects.....10

Where to find product documentation and other information12

Contacting NetApp Support.....13

Copyright.....14

Trademark.....15

How to send comments about documentation and receive update notifications.....16

Introduction

NetApp Element PowerShell Tools is a collection of PowerShell Core functions that use Element storage API to control an Element storage system. These functions allow administrators to query for information, make changes to objects in a storage system, and develop complex scripts on a single platform. You can use this module with other modules and snap-ins, such as VMware PowerCLI to extend capabilities throughout the infrastructure.

Any user with an Element storage system and PowerShell Core can take advantage of NetApp Element PowerShell Tools. Before you use NetApp Element PowerShell Tools, you should have an understanding of PowerShell Core functions.

You can download the NetApp Element PowerShell Tools Install guide and software from the NetApp Support [Site](#).

How to use NetApp Element PowerShell Tools

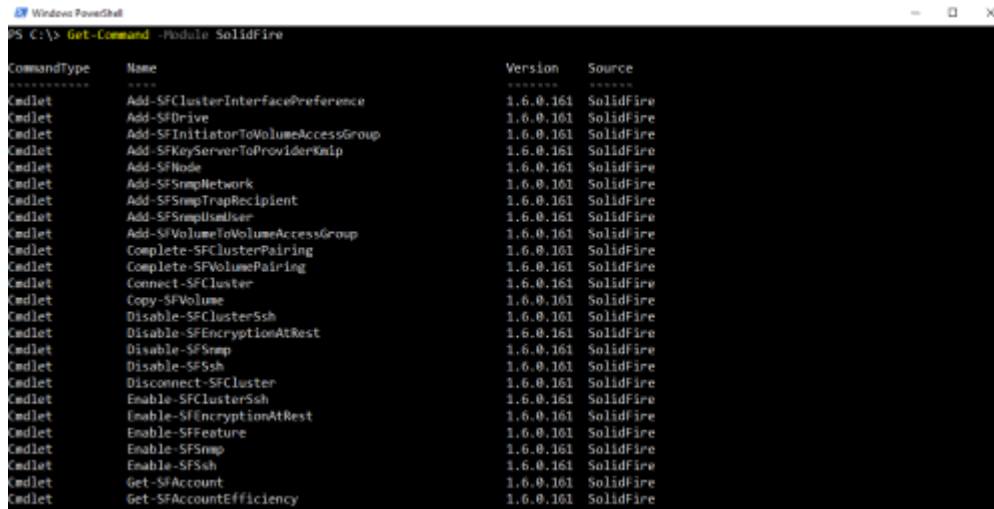
The following topics describe ways to access available functions for NetApp Element PowerShell Tools, manage connections to an Element storage node, and find additional cmdlet parameter and return object information.

Listing available functions

The available functions for NetApp Element PowerShell Tools can be explored using the native Get-Command PowerShell Tools cmdlet.

Steps

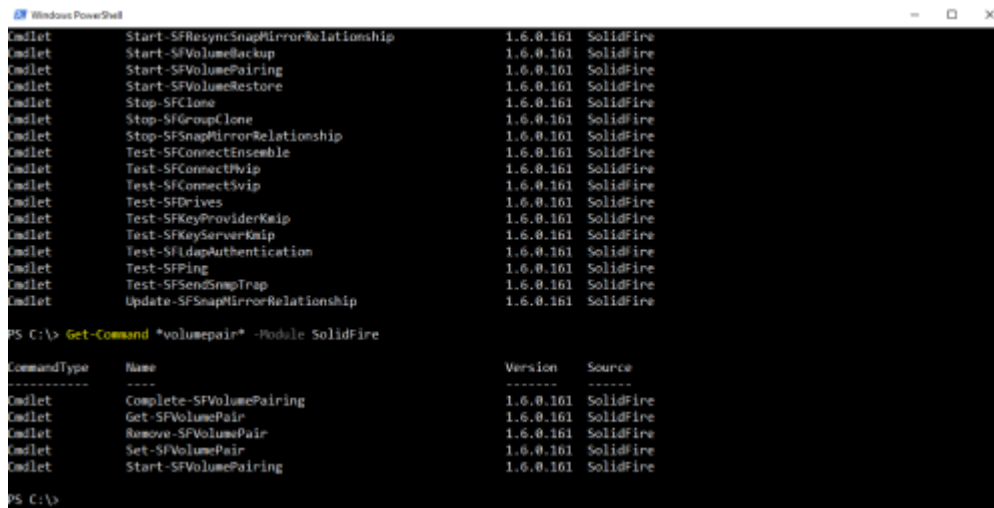
1. In the command line interface, type the following: `Get-Command -Module SolidFire`
The list of available commands appears.



```
PS C:\> Get-Command -Module SolidFire

CommandType      Name                                     Version      Source
-----
Cmdlet            Add-SFClusterInterfacePreference        1.6.0.161    SolidFire
Cmdlet            Add-SFDrive                             1.6.0.161    SolidFire
Cmdlet            Add-SFInitiatorToVolumeAccessGroup      1.6.0.161    SolidFire
Cmdlet            Add-SFKeyServerToProviderKmap           1.6.0.161    SolidFire
Cmdlet            Add-SFNode                              1.6.0.161    SolidFire
Cmdlet            Add-SFSnapNetwork                       1.6.0.161    SolidFire
Cmdlet            Add-SFSnapTrapRecipient                  1.6.0.161    SolidFire
Cmdlet            Add-SFSnapUser                           1.6.0.161    SolidFire
Cmdlet            Add-SFVolumeToVolumeAccessGroup          1.6.0.161    SolidFire
Cmdlet            Complete-SFClusterPairing                1.6.0.161    SolidFire
Cmdlet            Complete-SFVolumePairing                 1.6.0.161    SolidFire
Cmdlet            Connect-SFCluster                        1.6.0.161    SolidFire
Cmdlet            Copy-SFVolume                            1.6.0.161    SolidFire
Cmdlet            Disable-SFClusterSsh                     1.6.0.161    SolidFire
Cmdlet            Disable-SFEncryptionAtRest               1.6.0.161    SolidFire
Cmdlet            Disable-SFSnap                           1.6.0.161    SolidFire
Cmdlet            Disable-SFSsh                            1.6.0.161    SolidFire
Cmdlet            Disconnect-SFCluster                     1.6.0.161    SolidFire
Cmdlet            Enable-SFClusterSsh                      1.6.0.161    SolidFire
Cmdlet            Enable-SFEncryptionAtRest                1.6.0.161    SolidFire
Cmdlet            Enable-SFFeature                         1.6.0.161    SolidFire
Cmdlet            Enable-SFSnap                            1.6.0.161    SolidFire
Cmdlet            Enable-SFSsh                             1.6.0.161    SolidFire
Cmdlet            Get-SFAccount                            1.6.0.161    SolidFire
Cmdlet            Get-SFAccountEfficiency                  1.6.0.161    SolidFire
```

2. Type a search term with an asterisk before and after the term to filter the command list: `Get-Command *volumepair* -Module SolidFire`
The filtered list of available commands appears.



```
PS C:\> Get-Command *volumepair* -Module SolidFire

CommandType      Name                                     Version      Source
-----
Cmdlet            Complete-SFVolumePairing                  1.6.0.161    SolidFire
Cmdlet            Get-SFVolumePair                         1.6.0.161    SolidFire
Cmdlet            Remove-SFVolumePair                       1.6.0.161    SolidFire
Cmdlet            Set-SFVolumePair                         1.6.0.161    SolidFire
Cmdlet            Start-SFVolumePairing                     1.6.0.161    SolidFire
```

Accessing embedded help

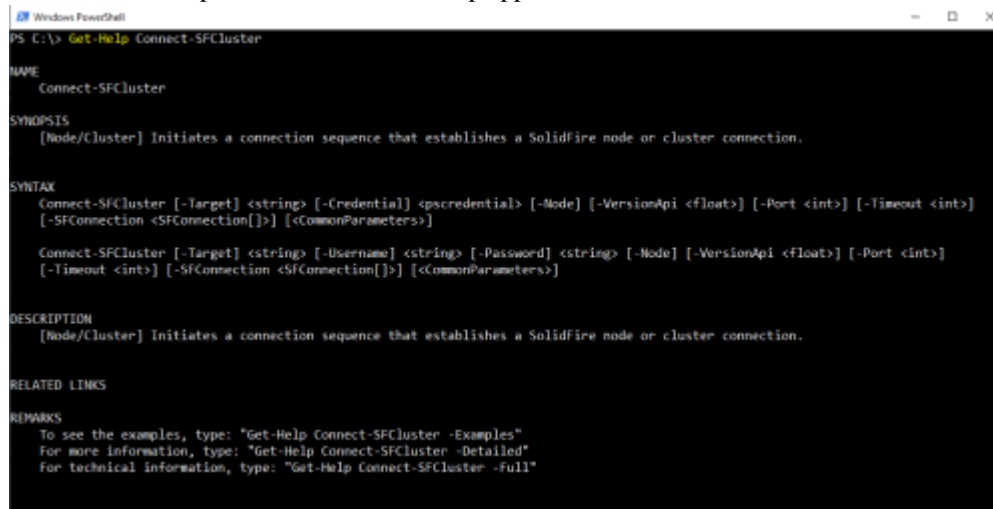
NetApp Element PowerShell Tools contains help examples that are accessible through the command line. Help content includes details about each command and examples of each function in use.

Step

In the command line interface, type

Get-Help <cmdlet>

The cmdlet description from embedded help appears.



```
PS C:\> Get-Help Connect-SFCluster

NAME
    Connect-SFCluster

SYNOPSIS
    [Node/Cluster] Initiates a connection sequence that establishes a SolidFire node or cluster connection.

SYNTAX
    Connect-SFCluster [-Target] <string> [-Credential] <pscredential> [-Mode] [-VersionApi <float>] [-Port <int>] [-Timeout <int>]
    [-SFConnection <SFConnection[]>] [<CommonParameters>]

    Connect-SFCluster [-Target] <string> [-Username] <string> [-Password] <string> [-Mode] [-VersionApi <float>] [-Port <int>]
    [-Timeout <int>] [-SFConnection <SFConnection[]>] [<CommonParameters>]

DESCRIPTION
    [Node/Cluster] Initiates a connection sequence that establishes a SolidFire node or cluster connection.

RELATED LINKS

REMARKS
    To see the examples, type: "Get-Help Connect-SFCluster -Examples"
    For more information, type: "Get-Help Connect-SFCluster -Detailed"
    For technical information, type: "Get-Help Connect-SFCluster -Full"
```

Note: To view full cmdlet help, see *Accessing return values using Get-Help*.

[Accessing return values using Get-Help](#) on page 9

Parameter sets

Many of the functions for NetApp Element PowerShell Tools have parameter sets to allow multiple use cases. For example, parameter sets are used with the creation and modification of Element storage objects, such as Accounts, Volumes, and Volume Access Groups.

You can identify parameter sets by using `Get-Help` for the function and reviewing the content under the Syntax section.

Managing connections to an Element storage cluster

All of the functions in NetApp Element PowerShell Tools make direct calls to the Element storage API. In order to manage authentication efficiently, a connection function has been developed for collecting target and authentication information.

Connecting to an Element storage cluster

Use `Connect-SFCluster` to connect to an Element storage cluster. The function collects Cluster connection, including target and authentication information from the user. `Connect-SFCluster` also supports connections to multiple Element storage clusters.

About this task

By default, the `Connect-SFCluster` queries the target cluster and sets the connection information to the latest API version on the cluster. This could also change the URI property to the version of Element software you are using. See *Changing API versions* to specify an API.

[Changing API versions](#) on page 7

Step

In the command line interface, type

`Connect-SFCluster -Target "<address>"`

The following example shows a successful connection.



Note: If your connection to an Element storage cluster is successful, the function `Connect-SFCluster` stores credentials and target information into a global variable `$SFConnection`. Multiple connections are also supported, and each successful connection is stored in the global array variable `$SFConnection`. See *Global variables for all functions*.

[Global variables for all functions](#) on page 8

Connecting to an Element storage node

Use the `Connect-SFCluster` function with a `-Node` switch parameter to connect to a specific Element storage node.

Steps

1. In the command line interface, type the following to get the IP address for the node:

```
Get-SFNode | Select Name, ManagementIP, NodeID
```

2. Include `-Node` and provide the node IP address in order to connect.

```
Connect-SFCluster -Target <NodeIP> -UserName <AdminAccount> -Node:
```

Disconnecting from an Element storage cluster or node

Use the `Disconnect-SFCluster` function to disconnect from an Element storage cluster. The function also clears the `$SFConnection` and `$SFConnections` global variables from the session. This makes it easier to secure the shell if you wish to keep it active or work with a different Element storage cluster.

About this task

You can disconnect a specific connection using the name of the connection from the `$SFConnection` or `$SFConnections` global variables. This name is either the cluster or the node name. See *Global variables for all functions* for an example.

[Global variables for all functions](#) on page 8

Step

In the command line interface, type the following to disconnect from the cluster:

Disconnect-SFCluster

Note: You can add an optional extension `-Name <node or cluster name>` or optional `-Target` parameter to specify the IP address instead of the name.

The following example shows a successful disconnection.



```
PS C:\> Disconnect-SFCluster -name <redacted> -FOR-TEST-ONLY
Disconnected from <redacted> - 10.117.0.1
PS C:\>
```

The screenshot shows a PowerShell console window with the following content:

```
Credential : System.Management.Automation.PSCredential
Limits : [{"AccountCountMax" = 5000, "AccountNameLengthMax" = 64, "AccountNameLengthMin" = 1, "BulkVolumeJobsPerNodeMax" = 8, "BulkVolumeJobsPerVolumeMax" = 2, "CloneJobsPerVolumeMax" = 2, "ClusterPairsCountMax" = 4, "InitiatorNameLengthMax" = 224, "InitiatorCountMax" = 10000, "InitiatorsPerVolumeAccessGroupCountMax" = 128, "IscsiSessionsFromFibreChannelNodesMax" = 4096, "QosPolicyCountMax" = 500, "SecretLengthMax" = 16, "ScheduleNameLengthMax" = 244, "SecretLengthMin" = 12, "SnapshotNameLengthMax" = 255, "SnapshotsPerVolumeMax" = 32, "VolumeAccessGroupCountMax" = 10000, "VolumeAccessGroupLunMax" = 16383, "VolumeAccessGroupNameLengthMax" = 64, "VolumeAccessGroupNameLengthMin" = 1, "VolumeAccessGroupsPerInitiatorCountMax" = 1, "VolumeAccessGroupsPerVolumeCountMax" = 64, "InitiatorAliasLengthMax" = 224, "VolumeBurstIOPSMax" = 200000, "VolumeBurstIOPSMin" = 100, "VolumeCountMax" = 1000, "VolumeMaxIOPSMax" = 200000, "VolumeMaxIOPSMin" = 100, "VolumeMinIOPSMax" = 15000, "VolumeMinIOPSMin" = 50, "VolumeNameLengthMax" = 64, "VolumeNameLengthMin" = 1, "VolumeSizeMax" = 1759218604416, "VolumeSizeMin" = 1000000000, "VolumesPerAccountCountMax" = 2000, "VolumesPerGroupSnapshotMax" = 32, "VolumesPerVolumeAccessGroupCountMax" = 2000}]
Timeout :
Element : SolidFire.Element.Api.SolidFireElement

PS C:\> Disconnect-SFCluster -name <redacted> -FOR-TEST-ONLY
Disconnected from <redacted> - 10.117.0.1
PS C:\>
```

Changing API versions

Use `-VersionApi` to specify an Element storage API version. By default, the NetApp Element PowerShell Tools module will use the most recent version of the Element storage API available.

About this task

Note: Changing versions might produce unexpected results based on availability of features and possible API method changes between releases. Even if a connection to an API version works, not all new features might be available in the NetApp Element PowerShell Tools module version that you have installed.

Step

Type the following command into the command line to specify the API version:

Connect-SFCluster -Target <address> -VersionApi <version number>

The following example demonstrates connecting to an Element storage cluster with API version 8.0 (Oxygen).

```
Windows PowerShell
PS C:\> Connect-SFCluster -target 10.117.1.1 -user admin -password 123456 -versionAPI 8

Target      : 10.117.1.1
Name        : nt_container-FOR-TEST-ONLY
Port        :
VersionApiName : Oxygen
VersionApiNumber : 8.0
Node        : False
Uri         : https://10.117.1.1/json-rpc/8.0
RequestCount : 3
Credential  : System.Management.Automation.PSCredential
Limits      : {"AccountCountMax" = 5000, "AccountNameLengthMax" = 64, "AccountNameLengthMin" = 1, "BulkVolumeJobsPerNodeMax" = 8, "BulkVolumeJobsPerVolumeMax" = 2, "CloneJobsPerVolumeMax" = 2, "ClusterPairsCountMax" = 4, "InitiatorNameLengthMax" = 224, "InitiatorCountMax" = 10000, "InitiatorsPerVolumeAccessGroupCountMax" = 128, "IscsiSessionsFromFibreChannelNodesMax" = 4096, "QosPolicyCountMax" = 500, "SecretLengthMax" = 16, "ScheduleNameLengthMax" = 244, "SecretLengthMin" = 12, "SnapshotNameLengthMax" = 255, "SnapshotsPerVolumeMax" = 32, "VolumeAccessGroupCountMax" = 1000, "VolumeAccessGroupLunMax" = 16383, "VolumeAccessGroupNameLengthMax" = 64, "VolumeAccessGroupNameLengthMin" = 1, "VolumeAccessGroupsPerInitiatorCountMax" = 1, "VolumeAccessGroupsPerVolumeCountMax" = 64, "InitiatorAliasLengthMax" = 224, "VolumeBurstIOPSMax" = 200000, "VolumeBurstIOPSMIn" = 100, "VolumeCountMax" = 1000, "VolumeMaxIOPSMax" = 200000, "VolumeMaxIOPSMIn" = 100, "VolumeMinIOPSMax" = 15000, "VolumeMinIOPSMIn" = 50, "VolumeNameLengthMax" = 64, "VolumeNameLengthMin" = 1, "VolumeSizeMax" = 17592186044416, "VolumeSizeMin" = 1000000000, "VolumesPerAccountCountMax" = 2000, "VolumesPerGroupSnapshotMax" = 32, "VolumesPerVolumeAccessGroupCountMax" = 2000}
Timeout     :
Element     : SolidFire.Element.Api.SolidFireElement

PS C:\>
```

Global variables for all functions

If your connection to an Element storage API cluster is successful, the function `Connect-SFCluster` stores credentials and target information in a global variable `$SFConnection`.

The information in this variable is used in API calls of other NetApp Element PowerShell Tools functions. Multiple connections are also supported, and each successful connection is stored in the global array variable `$SFConnections`.

```
Windows PowerShell
PS C:\> $SFConnections

Target      : 10.117.1.1
Name        : nt_container-FOR-TEST-ONLY
Port        :
VersionApiName : Oxygen
VersionApiNumber : 8.0
Node        : False
Uri         : https://10.117.1.1/json-rpc/8.0
RequestCount : 3
Credential  : System.Management.Automation.PSCredential
Limits      : {"AccountCountMax" = 5000, "AccountNameLengthMax" = 64, "AccountNameLengthMin" = 1, "BulkVolumeJobsPerNodeMax" = 8, "BulkVolumeJobsPerVolumeMax" = 2, "CloneJobsPerVolumeMax" = 2, "ClusterPairsCountMax" = 4, "InitiatorNameLengthMax" = 224, "InitiatorCountMax" = 10000, "InitiatorsPerVolumeAccessGroupCountMax" = 128, "IscsiSessionsFromFibreChannelNodesMax" = 4096, "QosPolicyCountMax" = 500, "SecretLengthMax" = 16, "ScheduleNameLengthMax" = 244, "SecretLengthMin" = 12, "SnapshotNameLengthMax" = 255, "SnapshotsPerVolumeMax" = 32, "VolumeAccessGroupCountMax" = 1000, "VolumeAccessGroupLunMax" = 16383, "VolumeAccessGroupNameLengthMax" = 64, "VolumeAccessGroupNameLengthMin" = 1, "VolumeAccessGroupsPerInitiatorCountMax" = 1, "VolumeAccessGroupsPerVolumeCountMax" = 64, "InitiatorAliasLengthMax" = 224, "VolumeBurstIOPSMax" = 200000, "VolumeBurstIOPSMIn" = 100, "VolumeCountMax" = 1000, "VolumeMaxIOPSMax" = 200000, "VolumeMaxIOPSMIn" = 100, "VolumeMinIOPSMax" = 15000, "VolumeMinIOPSMIn" = 50, "VolumeNameLengthMax" = 64, "VolumeNameLengthMin" = 1, "VolumeSizeMax" = 17592186044416, "VolumeSizeMin" = 1000000000, "VolumesPerAccountCountMax" = 2000, "VolumesPerGroupSnapshotMax" = 32, "VolumesPerVolumeAccessGroupCountMax" = 2000}
Timeout     :
Element     : SolidFire.Element.Api.SolidFireElement

Target      : 10.117.1.1
Name        : nt_container-FOR-TEST-ONLY
Port        :
VersionApiName : Oxygen
VersionApiNumber : 8.0
Node        : False
Uri         : https://10.117.1.1/json-rpc/8.0
RequestCount : 3
Credential  : System.Management.Automation.PSCredential
Limits      : {"AccountCountMax" = 5000, "AccountNameLengthMax" = 64, "AccountNameLengthMin" = 1, "BulkVolumeJobsPerNodeMax" = 8, "BulkVolumeJobsPerVolumeMax" = 2, "CloneJobsPerVolumeMax" = 2, "ClusterPairsCountMax" = 4, "InitiatorNameLengthMax" = 224, "InitiatorCountMax" = 10000, "InitiatorsPerVolumeAccessGroupCountMax" = 128, "IscsiSessionsFromFibreChannelNodesMax" = 4096, "QosPolicyCountMax" = 500, "SecretLengthMax" = 16, "ScheduleNameLengthMax" = 244, "SecretLengthMin" = 12, "SnapshotNameLengthMax" = 255, "SnapshotsPerVolumeMax" = 32, "VolumeAccessGroupCountMax" = 1000, "VolumeAccessGroupLunMax" = 16383, "VolumeAccessGroupNameLengthMax" = 64, "VolumeAccessGroupNameLengthMin" = 1, "VolumeAccessGroupsPerInitiatorCountMax" = 1, "VolumeAccessGroupsPerVolumeCountMax" = 64, "InitiatorAliasLengthMax" = 224, "VolumeBurstIOPSMax" = 200000, "VolumeBurstIOPSMIn" = 100, "VolumeCountMax" = 1000, "VolumeMaxIOPSMax" = 200000, "VolumeMaxIOPSMIn" = 100, "VolumeMinIOPSMax" = 15000, "VolumeMinIOPSMIn" = 50, "VolumeNameLengthMax" = 64, "VolumeNameLengthMin" = 1, "VolumeSizeMax" = 17592186044416, "VolumeSizeMin" = 1000000000, "VolumesPerAccountCountMax" = 2000, "VolumesPerGroupSnapshotMax" = 32, "VolumesPerVolumeAccessGroupCountMax" = 2000}
Timeout     :
Element     : SolidFire.Element.Api.SolidFireElement

PS C:\>
```


Common parameters

All cmdlets, with the exception of `Connect-SFCluster` and `Disconnect-SFCluster`, have the common parameters `Target` and `SFConnection`. These common parameters are not in the `Get-Help` examples for each cmdlet but can be assumed to be present. The embedded Help within NetApp Element PowerShell Tools has the common parameter in its examples.

If the `Target` parameter is included in the cmdlet, the cmdlet will run against all connections in `$SFConnections` whose name or target (IP address) matches using a wildcard pattern match. Results are written to output without indicating the target against which the cmdlet was run.

If the `-SFConnection` parameter is included in the cmdlet, the cmdlet will run against the specific `SFConnection` that was handed in through that parameter. You can inspect the `$SFConnections` session variable to find a specific `SFConnection` or you can pass the result of `Connect-SFCluster` into it.

Each cmdlet is configured to be run on either a cluster or node. Before processing the cmdlet against any target, the cmdlet will check the connection to make sure it matches the intended cluster or node. If there is no match, a non-terminating error message (as in the following example) appears that states it is skipping the command:

```
Get-SFNetworkConfig : Skipping command on connection 'Connection Name'. Cmdlet requires  
Node connection.
```

All cmdlets will execute against all matching connections.

Return object descriptions

Return values are fully documented as part of the .NET SDK documentation that is available online.

There are three methods for inspecting cmdlet return values:

- [Accessing return value reference documentation](#) on page 9
- [Accessing return values using Get-Help](#) on page 9
- [Leveraging Get-Member to inspect return objects](#) on page 10

Accessing return value reference documentation

Each return value is documented as part of the online documentation for the Element .NET SDK.

This documentation can be found on the [NetApp Support site](#).

Accessing return values using Get-Help

For any cmdlet included in NetApp Element PowerShell Tools, type `Get-Help <cmdlet name> -Full` to return command details.

- A specific return type for the cmdlet that is described in the Outputs section.
- A URL to the related Element .NET SDK reference page on GitHub.
- Examples of cmdlet use.

The following is an example of `Get-Help <cmdlet name> -Full`:

```
Windows PowerShell
PS C:\> Get-Help get-sfvolume -Full

NAME
    Get-SFVolume

SYNOPSIS
    [Cluster] Gets a list of volumes from the cluster.

SYNTAX
    Get-SFVolume [[-VolumeID <long[]>] [[-ExcludeVWOLs]] [[-IncludeDeleted]] [-Target <string[]>] [-SFConnection
    <SFConnection[]>] [<CommonParameters>]

    Get-SFVolume [[-Name <string[]>] [[-ExcludeVWOLs]] [[-IncludeDeleted]] [-Target <string[]>] [-SFConnection <SFConnection[]>]
    [<CommonParameters>]

    Get-SFVolume [-AccountID <long[]>] [[-ExcludeVWOLs]] [[-IncludeDeleted]] [-Target <string[]>] [-SFConnection <SFConnection[]>]
    [<CommonParameters>]

    Get-SFVolume [-Account <Account[]>] [[-ExcludeVWOLs]] [[-IncludeDeleted]] [-Target <string[]>] [-SFConnection
    <SFConnection[]>] [<CommonParameters>]

DESCRIPTION
    [Cluster] Gets a list of volumes from the cluster.

PARAMETERS
    -Account <Account[]>
        Specify account(s).

        Required?                true
        Position?                 0
        Default value
        Accept pipeline input?    true (ByValue, ByPropertyName)
        Accept wildcard characters?

    -AccountID <long[]>
        Enter an Account ID or list of Account IDs.

        Required?                true
        Position?                 0
        Default value
        Accept pipeline input?    true (ByValue, ByPropertyName)
        Accept wildcard characters?

    -ExcludeVWOLs
        Use -ExcludeVWOLs to exclude volumes that are Virtual Volumes from the returned volumes.

        Required?                false
        Position?                 1
        Default value
        Accept pipeline input?    false
        Accept wildcard characters?
```

Leveraging Get-Member to inspect return objects

Use the built-in NetApp Element PowerShell Tools `Get-Member` cmdlet to inspect return values.

Steps

1. In the command line interface, type:

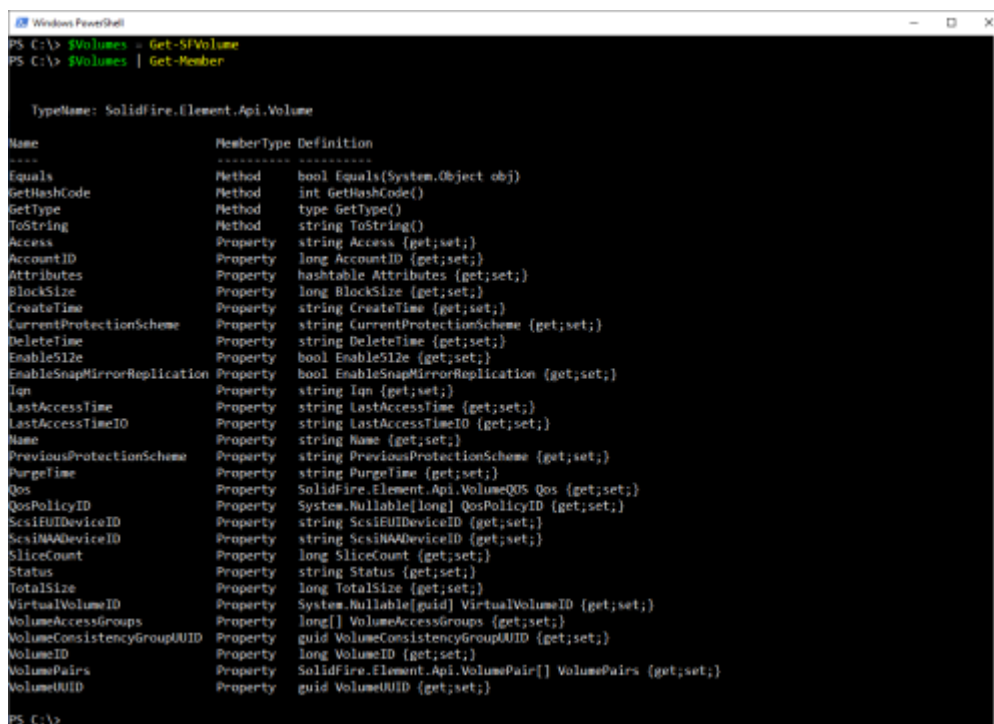
```
$<variable> = <Cmdlet with appropriate parameters>
```

2. Type:

```
$<variable> | Get-Member
```

The following example shows the result for `Get-SFVolume` with each return property listed.

How to use NetApp Element PowerShell Tools



```

Windows PowerShell
PS C:\> $Volumes = Get-SFVolume
PS C:\> $Volumes | Get-Member

TypeName: SolidFire.Element.Api.Volume

Name                MemberType Definition
-----
Equals              Method      bool Equals(System.Object obj)
GetHashCode          Method      int GetHashCode()
GetType             Method      type GetType()
ToString            Method      string ToString()
Access              Property    string Access {get;set;}
AccountID           Property    long AccountID {get;set;}
Attributes           Property    hashtable Attributes {get;set;}
BlockSize           Property    long BlockSize {get;set;}
CreateTime          Property    string CreateTime {get;set;}
CurrentProtectionScheme Property    string CurrentProtectionScheme {get;set;}
DeleteTime          Property    string DeleteTime {get;set;}
EnableS12e          Property    bool EnableS12e {get;set;}
EnableSnapMirrorReplication Property    bool EnableSnapMirrorReplication {get;set;}
Iqn                 Property    string Iqn {get;set;}
LastAccessTime      Property    string LastAccessTime {get;set;}
LastAccessTimeIO    Property    string LastAccessTimeIO {get;set;}
Name                Property    string Name {get;set;}
PreviousProtectionScheme Property    string PreviousProtectionScheme {get;set;}
PurgeTime           Property    string PurgeTime {get;set;}
Qos                 Property    SolidFire.Element.Api.VolumeQos {get;set;}
QosPolicyID         Property    System.Nullable[long] QosPolicyID {get;set;}
ScsiFUIDeviceID     Property    string ScsiFUIDeviceID {get;set;}
ScsiHAAADeviceID    Property    string ScsiHAAADeviceID {get;set;}
SliceCount          Property    long SliceCount {get;set;}
Status              Property    string Status {get;set;}
TotalSize           Property    long TotalSize {get;set;}
VirtualVolumeID     Property    System.Nullable[guid] VirtualVolumeID {get;set;}
VolumeAccessGroups  Property    long[] VolumeAccessGroups {get;set;}
VolumeConsistencyGroupUUID Property    guid VolumeConsistencyGroupUUID {get;set;}
VolumeID            Property    long VolumeID {get;set;}
VolumePairs         Property    SolidFire.Element.Api.VolumePair[] VolumePairs {get;set;}
VolumeUUID          Property    guid VolumeUUID {get;set;}
PS C:\>

```

3. If objects are more than one layer deep (see the QoS property in the example from the previous step), examine additional layers using the dot operator:

`$<variable.property> | Get-Member`

Where to find product documentation and other information

You can learn more about using and managing NetApp HCI and SolidFire all-flash storage from the resources available in the Documentation Centers and Resources pages for both products.

In the Documentation Centers, you can also find information about hardware installation and maintenance, additional content resources available, links to known issues and resolved issues, and the latest release notes. On the Resources pages, you can find links to data sheets, technical reports, white papers, and videos.

- [*NetApp HCI Documentation Center*](#)
- [*NetApp HCI Resources page*](#)
- [*SolidFire and Element 11.8 Documentation Center*](#)
- [*SolidFire and Element 11.7 Documentation Center*](#)
- [*SolidFire and Element 11.5 Documentation Center*](#)
- [*SolidFire and Element 11.3 Documentation Center*](#)
- [*SolidFire Resources page*](#)

Contacting NetApp Support

If you need help with or have questions or comments about NetApp products, contact NetApp Support.

- Web:
mysupport.netapp.com
- Phone:
 - 888.4.NETAPP (888.463.8277) (US and Canada)
 - 00.800.44.638277 (EMEA/Europe)
 - +800.800.80.800 (Asia/Pacific)

Copyright

Copyright © 2020 NetApp, Inc. All rights reserved. Printed in the U.S.

No part of this document covered by copyright may be reproduced in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system—without prior written permission of the copyright owner.

Software derived from copyrighted NetApp material is subject to the following license and disclaimer:

THIS SOFTWARE IS PROVIDED BY NETAPP "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL NETAPP BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of NetApp.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

Data contained herein pertains to a commercial item (as defined in FAR 2.101) and is proprietary to NetApp, Inc. The U.S. Government has a non-exclusive, non-transferrable, non-sublicensable, worldwide, limited irrevocable license to use the Data only in connection with and in support of the U.S. Government contract under which the Data was delivered. Except as provided herein, the Data may not be used, disclosed, reproduced, modified, performed, or displayed without the prior written approval of NetApp, Inc. United States Government license rights for the Department of Defense are limited to those rights identified in DFARS clause 252.227-7015(b).

Trademark

NETAPP, the NETAPP logo, and the marks listed on the NetApp Trademarks page are trademarks of NetApp, Inc. Other company and product names may be trademarks of their respective owners.

<http://www.netapp.com/us/legal/netapptmlist.aspx>

How to send comments about documentation and receive update notifications

You can help us to improve the quality of our documentation by sending us your feedback. You can receive automatic notification when production-level (GA/FCS) documentation is initially released or important changes are made to existing production-level documents.

If you have suggestions for improving this document, send us your comments by email.

doccomments@netapp.com

To help us direct your comments to the correct division, include in the subject line the product name, version, and operating system.

If you want to be notified automatically when production-level documentation is released or important changes are made to existing production-level documents, follow Twitter account @NetAppDoc.

You can also contact us in the following ways:

- NetApp, Inc., 1395 Crossman Ave., Sunnyvale, CA 94089 U.S.
- Telephone: +1 (408) 822-6000
- Fax: +1 (408) 822-4501
- Support telephone: +1 (888) 463-8277