OnCommand® Plug-in 4.1 for Microsoft®

Windows PowerShell Cmdlet and Orchestrator Activity Reference Guide

NetApp, Inc.
495 East Java Drive
Sunnyvale, CA 94089
U.S.

Telephone: +1 (408) 822-6000
Fax: +1 (408) 822-4501
Support telephone: +1 (888) 463-8277
Web: www.netapp.com
Feedback: doccomments@netapp.com

Part number: 215-09254_A0
November 2014

Contents

Windows PowerShell execution policies and administration profiles

To be able to run cmdlets from Windows PowerShell, you must first set execution policies to the correct levels and create administrator profiles on your local host.

PowerShell execution policy values

You must set the proper execution policies to run cmdlets in the plug-in. The following list includes the possible values for Windows PowerShell execution policies:

AllSigned

Allows scripts that have been verified with security certificates to run. To run the cmdlets securely, you should set the Windows PowerShell execution policy to AllSigned.

Restricted

Restricts any scripts from being run, downloaded, or installed. Restricted is the default policy.

RemoteSigned

Allows scripts to be run, downloaded, or installed. This policy is not secure, and malicious code can easily exploit and damage any PowerShell .ps1 or VB script files.

Bypass

Nothing is blocked and there are no warnings or prompts.

Undefined

There is no execution policy set in the current scope.

PowerShell Administration Profiles

You must import the OC.Cmdlets module to your local Windows PowerShell profile if you want the cmdlets to be available from any Windows PowerShell session. This enables you to use the OC.Cmdlets module through PowerShell sessions from Microsoft applications, such as System Center Virtual Machine Manager (SCVMM) and System Center Operations Manager (SCOM).

You can run the following command from Windows PowerShell to get complete information about execution policies:

```
get-help about_execution_policies
```

For more information about how to set Windows PowerShell execution policies, see the Microsoft TechNet web site.

Common cmdlet parameters

The Windows PowerShell cmdlets include both common cmdlet parameters and risk-mitigation parameters that you can use to customize the operation that the cmdlet performs.

Cmdlets and parameters are not case-sensitive.

Cmdlet parameters

You can use the following parameters with all cmdlets:

`[-Debug {True | False}]`

Displays information about the operation.

`[-ErrorAction {SilentlyContinue | Continue | Inquire | Stop}]`

Determines how the cmdlet responds to a warning when performing the operation. The following list describes what each value means:

SilentlyContinue

Suppresses the warning message and continues the operation.

Continue

Displays the warning message and continues the operation. This is the default value for this parameter.

Inquire

Displays the warning message and asks if you want to continue the operation.

Stop

Displays the warning message and stops the operation.

`-ErrorVariable |` ***`Variable_name`***

Stores errors about the cmdlet in the specified variable.

`-OutBuffer |` ***`Object_number`***

Determines the number of objects that can reside in the buffer before they are sent.

`-OutVariable |` ***`Variable_name`***

Displays objects output by the cmdlet and then stores them in the specified variable.

`[-Verbose {True | False}]`

Displays detailed information about the operation.

`[-WarningAction {SilentlyContinue | Continue | Inquire | Stop}]`

Determines how the cmdlet responds to a warning when performing an operation. The following list describes what each value means:

SilentlyContinue

Suppresses the warning message and continues the operation.

Continue

Displays the warning message and continues the operation. This is the default value for this parameter.

Inquire

Displays the warning message and asks if you want to continue.

Stop

Displays the warning message and stops the operation.

`-WarningVariable` | ***`Variable_name`***

Stores warnings about the cmdlet in the specified variable.

Risk mitigation parameters

You can add the following parameters to cmdlets to request confirmation before cmdlets perform the specified action:

`[-Confirm {True | False}]`

Prompts you for input before executing the operation.

`[-WhatIf {True | False}]`

Displays a message that describes the effects of running the cmdlet before that action is performed.

Table of PowerShell cmdlets and corresponding Orchestrator activities

Several PowerShell cmdlets have corresponding Orchestrator activities. You can use this table to determine which activities to use in place of a particular cmdlet if you choose to use Orchestrator rather than PowerShell or vice versa.

Storage System cache credential cmdlets and activities

The following table lists the Storage System cache credential cmdlets and corresponding activities:

PowerShell cmdlet	Orchestrator activity
Add-OCStorageSystem	Add Controller Credentials
Get-OCStorageSystem	List Controller Credentials
Remove-OCStorageSystem	Remove Controller Credentials

Provisioning cmdlets and activities

The following table lists the provisioning cmdlets and corresponding activities:

PowerShell cmdlet	Orchestrator activity
Connect-OCStorage	Connect Storage to Host
Disconnect-OCStorage	Disconnect Storage
Get-OCStorage	List Storage
New-OCStorage	Provision Storage
Remove-OCStorage	Delete Storage
Set-OCStorageSize	Resize Storage
Remove-OCVM	Delete VM on Host

Cloning cmdlets and activities

The following table lists the cloning cmdlets and corresponding activities:

PowerShell cmdlet	Orchestrator activity
New-OCClone	Clone VM; Clone VM from SCVMM Template
New-OCCloneFile	Clone NTFS File

Disaster recovery cmdlets and activities

The following table lists the disaster recovery cmdlets and corresponding activities:

PowerShell cmdlet	Orchestrator activity
New-OCDRPlan	Create DR Plan
Confirm-OCDRPlan	Validate DR Plan
Update-OCDRPlan	Update a DR Plan
Connect-OCDRLUN	Connect DR Storage
Initialize-OCDRMirror	Initialize SnapMirror
Invoke-OCDRMirrorStop	Abort SnapMirror
Invoke-OCDRMirrorBreak	Break SnapMirror
Invoke-OCDRMirrorQuiesce	Quiesce SnapMirror
Invoke-OCDRMirrorResume	Resume SnapMirror
Invoke-OCDRMirrorResync	Resync SnapMirror
Invoke-OCDRMirrorReverseResync	Reverse Resync SnapMirror
Invoke-OCDRMirrorUpdate	Update SnapMirror
Get-OCDRMirrorStatus	Get SnapMirror Status
Reset-OCDRSite	Clean-up a DR Site
Restore-OCDRVM	Restore VMs
Invoke-OCDRMirrorClear	Release SnapMirror
Invoke-OCDRFailover	DR Failover
Invoke-OCDRFailback	DR Failback
Start-OCDRSimulateFailover	Start Test DR Failover
Stop-OCDRSimulateFailover	End Test DR Failover

Debug-OCHost cmdlet

The `Debug-OCHost` cmdlet runs diagnostics on the local machine and monitored storage systems, which enables you to check the host configurations on your storage systems. When you encounter issues with the plug-in, you can run the `Debug-OCHost` cmdlet to help you resolve those issues.

Syntax

```
Debug-OCHost
[- ManagementPackRules{ALL|PRO|STORAGE|VIRTUALIZATION|CLUSTER|
CLUSTERVIRTUALIZATION|METROCLUSTER}]
[- CheckVersion{ALL|CLUSTER|METROCLUSTER}]
[- CheckSettings, settings, s}]
[- CheckAllowedLicenses}]
[- CheckAllowedONTAPI}]
[- CheckDataONTAPManagementPacks}]
[- CheckManagementServerResourcePool}]
[- CheckMetroClusterStatus}]
[- CheckManagementServerMemory}]
[- ALL}]
[<SwitchParameter>]]
[<CommonParameters>]
```

Description

The following list shows supported configurations for this cmdlet:

OCPM version

3.2 or later required for all OCPM features

PowerShell version

3.0 or later required for all OCPM features

.NET Framework version

Microsoft .NET Framework 4.0 Full or later

Operating System version

Windows Server 2008 R2 SP1 or later

SCOM version

For SCOM management packs, Microsoft SCOM 2012 or later

Is SCOM Management server

For SCOM management server, True

SCVMM version

For SCVMM components, Microsoft SCVMM 2012 or later

Is SCVMM server

For SCVMM components, True

Is SCO 2012 Management server

For SCO components, True

Is SCO 2012 Runbook server

For SCO components, True

Clustered Data ONTAP version

Clustered Data ONTAP 8.2 or later

The following table provides first steps for troubleshooting:

Troubleshooting area	Problem	Recommended action
OCPM VIM service status	Not installed	Install the plug-in.
OCPM WMI provider status	Error	Install OCPM OnCommand Discovery Agent feature.
Remote Registry status	Not Configured	Configure Remote Registry required by OCPM SCVMM console add-ins: **1.** Open the Services console. **2.** Right-click **Remote Registry** and select **Properties**. **3.** In the Startup Type field, choose **Automatic**. **4.** Click **Apply**. **5.** Start the Remote Registry service.
Remote Powershell status	Disabled	Run the `Enable-PSRemoting` cmdlet to configure Remote Powershell.
Firewall settings	Not Configured	Run the `Enable-NetFirewallRule -DisplayGroup "Remote Volume Management"` cmdlet to configure the firewall settings.

Troubleshooting area	Problem	Recommended action
Multipath I/O	Disabled	Enable the Multipath I/O Windows feature. For more information, see the Microsoft TechNet web site.
Microsoft DSM	Not Configured	Run the `New-MSDSMSupportedHW -VendorId "NETAPP" -ProductId "LUN"` cmdlet to configure Microsoft DSM.
Data ONTAP DSM	Not Configured	Download and install Data ONTAP DSM. For more information, see *mysupport.netapp.com*.

Parameters

`-ManagementPackRules`

Lists SCOM management packs rules and their important parameters for the management packs specified. Valid values are as follows:

All

(Default) Gets management pack rule values for all OnCommand Data ONTAP management packs.

PRO

Gets management pack rule values for the OnCommand Data ONTAP PRO management pack.

Storage

Gets management pack rule values for the OnCommand Data ONTAP management pack.

Virtualization

Gets management pack rules values for the OnCommand Data ONTAP Virtualization management pack.

Cluster

Gets management pack rule values for the clustered Data ONTAP management pack.

Cluster virtualization

Gets management pack rules for the clustered Data ONTAP Virtualization management pack.

MetroCluster

Gets management pack rules for the clustered Data ONTAP MetroCluster management pack.

Required?

False

Position

0

Default value

ALL

Accept pipeline input?

False

Accept wildcard characters?

False

`-CheckVersion`

Checks the versions of the applications that are installed on the local server and monitored cluster storage systems (7-Mode storage systems are ignored by CheckVersion).

- **ALL:** (Default) Checks the versions of both the applications installed on the local server and monitored clustered storage systems.
- **CLUSTER:** Checks the versions of the connected clustered storage systems.
- **METROCLUSTER:** Checks the versions of the connected MetroCluster storage systems.

Required?

False

Position

1

Default value

ALL

Accept pipeline input?

False

Accept wildcard characters?

False

`-CheckSettings`

Checks the settings of the local server.

Required?

False

Position

2

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-CheckAllowedLicenses`

Checks the allowed licenses of the monitored clusters.

Required?

False

Position

3

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-CheckAllowedONTAPI`

Checks the allowed ONTAPIs library of the monitored storage systems.

Required?

False

Position

4

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-CheckDataONTAPManagementPacks`

Checks for the Data ONTAP management packs and determines if they are installed on the local server.

Required?

False

Position

5

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-CheckManagementServerResourcePool`

Checks if the management servers in the local management group are in the OCPM resource pool and discovers the plug-in version that is installed.

Required?

False

Position

6

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-CheckMetroClusterStatus`

Displays the status of each MetroCluster cluster.

Required?

False

Position

7

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-CheckManagementServerMemory`

Checks if the local management server meets Microsoft minimum physical memory recommendation. If the data warehouse server is on the same host as the management server, the minimum memory recommendation is 16 GB; otherwise, it is 8 GB.

Required?

False

Position

8

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-ALL`

Selects all possible parameters.

Required?

False

Position

7

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

Example: Checking versions

```
C:\PS>Debug-OCHost -CheckVersion ALL -Verbose


VERBOSE: Starting Debug-OCHost
VERBOSE: Entering ProcessRecord Debug-OCHost
VERBOSE: WMI Query for OS version
VERBOSE: WMI Query for .Net version
```

```
OCPM Version : 4.1
PowerShell Version : 3.0
.NET Framework Version : .NET 4.0 Full, .NET 3.5 Service Pack 1
Operating System Version : Microsoft Windows Server 2012 Standard
SCOM Version : System Center Operations Manager 2012 Server
SCOM Version Number : 7.0.9538.0
Is SCOM Management Server? : True
SCVMM Version : 3.1.3612.0
Is SCVMM Server? : True
Is SCO 2012 Management Server? : True
Is SCO 2012 Runbook Server? : True

VERBOSE: Getting monitored Cluster and SVMs
IPAddress : 10.225.12.154
HostName : mva-f18-3170-1_2
DataONTAPVersion : ClusteredDataONTAPVersion Release 8.2.0

IPAddress : 10.225.13.178
HostName : mva-bach-4-d1
DataONTAPVersion : ClusteredDataONTAPVersion Release 8.2.1

VERBOSE: Completed Debug-OCHost
```

Example: Checking settings

```
PS C:\Users\Administrator.MVA> Debug-OCHost -CheckSettings -Verbose


VERBOSE: Starting Debug-OCHost
VERBOSE: Entering ProcessRecord Debug-OCHost
VERBOSE: WMI Query for OS version
VERBOSE: Checking VIM web service status
VERBOSE: Checking VIM web service port number
VERBOSE: Checking Remote Registry status
VERBOSE: Checking WMI provider status status
VERBOSE: Checking Remote Powershell status
VERBOSE: Checking Firewall settings
VERBOSE: Checking MPIO and DSM
VERBOSE: Check if SCOM Management Server
VERBOSE: Checking for SQL server instance name

SCOM Clustered Data ONTAP storage discovery: Enabled
SCOM Clustered Data ONTAP virtualization discovery: Enabled
SCOM 7-mode Data ONTAP storage discovery: Disabled
SCOM 7-mode Data ONTAP virtualization discovery: Disabled
OCPM VIM service status: Running
OCPM VIM service port number: 808
WMI provider status: OK
Remote Registry status: Configured
Remote powershell status: Enabled
Firewall settings: Configured
Multipath I/O: Enabled
Data ONTAP DSM: Not Configured
Microsoft DSM: Configured
SQL server instance name: MVA-S70\sql
SQL server port number: 1433

VERBOSE: Completed Debug-OCHost
```

Example: Checking Data ONTAP PRO management pack rules

```
PS $secpasswd = ConvertTo-SecureString "PlainTextPassword" -
AsPlainText -Force $mycreds = New-Object
System.Management.Automation.PSCredential ("username", $secpasswd)
Debug-OCHost -ManagementPackRules PRO -SCOMManagementServer
remoteServer12 -SCOMCredential $mycreds


RuleName: Data ONTAP PRO: Collect Hyper-V Host Events Rule
ManagementPack: OnCommand Data ONTAP PRO
RuleEnabled: True
DefaultRuleInterval: N/A
OverrideRuleInterval: N/A
RuleName: Data ONTAP PRO: Discovery Rule
ManagementPack: OnCommand Data ONTAP PRO
RuleEnabled: True
DefaultRuleInterval: N/A
OverrideRuleInterval: N/A
```

Example: Checking allowed licenses

```
PS Debug-OCHost -CheckAllowedLicenses -Verbose


VERBOSE: Starting Debug-OCHost
VERBOSE: Entering ProcessRecord Debug-OCHost
VERBOSE: Getting monitored Cluster and SVMs

IPAddress StorageSystemType AllowedLicenses
```

```
--------- ---------------- ---------------
10.225.12.154 Cluster {Cluster Base License,
CIFS License,...
10.225.13.180 Cluster {Cluster Base License,
NFS License, ...
VERBOSE: Completed Debug-OCHost
```

Example: Checking allowed licenses using the `Select-Object -ExpandProperty` command

```
PS Debug-OCHost -CheckAllowedLicenses -Verbose | Select-Object -
ExpandProperty AllowedLicenses


VERBOSE: Starting Debug-OCHost
VERBOSE: Entering ProcessRecord Debug-OCHost
VERBOSE: Getting monitored Cluster and SVMs
Cluster Base License
CIFS License
iSCSI License
SnapRestore License
SnapMirror License
iSCSI License
SnapRestore License
SnapMirror License

Cluster Base License
NFS License
CIFS License
iSCSI License
FCP License
FlexClone License

VERBOSE: Completed Debug-OCHost
```

What Data ONTAP Toolkit activities are

Data ONTAP Toolkit 3.1 activities enable you to invoke Data ONTAP commands from System Center Orchestrator. You can also use the activities individually, as an alternative to using the Data ONTAP command-line interface and PowerShell scripting workflows. The Data ONTAP Toolkit 3.1 is supported only for Data ONTAP 7-Mode.

Create a LUN by Size activity

You use the **Create a LUN by Size** activity to create a new LUN of a size that you specify, although the size of the LUN that you create might be larger than the size that you specify.

Keep the following information in mind when you use the **Create a LUN by Size** activity:

- The LUN is created at the location of the LUN path that you specify, and that location must not already contain any files.
- For better performance, you should create the LUN on the LUN OSType that corresponds to the host that the LUN is mapped to.
- The directory that you specify in the LUN path must be a qtree root directory.
  The LUN path can be in either of two formats:
  - `/vol/`*`volName/LUNName`* for a volume
  - `/vol/`*`volName/qtreeName/LUNName`* for a qtree

Properties

The following table indicates the information that you can enter in the properties tab for the **Create a LUN by Size** activity:

Property name	Type	Required?	Default
Controller name [1]	String	Required	
LUN path [2]	String	Required	
Size	Integer	Required	1
Unit type [3]	Enumerated	Required	GB
Type [4]	Enumerated OS Type	Optional	
Unreserved	Boolean	Optional	

1. Can be a host name or an IP address
2. `/vol/volName/LUNName`
3. KB, MB, GB, and TB
4. Operating system types and multiprotocol operating system types. The LUN Multiprotocol Type, or operating system type, specifies the operating system of the host that is accessing the LUN. It also determines the layout of data on the LUN, the geometry used to access that data, and the minimum and maximum size of the LUN.

    - AIX: The host operating system is AIX.
    - HPUX: The host operating system is HPUX.
    - Hyper-V: You are using Windows Server 2008 Hyper-V and your LUNs contain virtual hard disks (VHDs). For raw LUNs, use the child operating system type as the LUN Multiprotocol Type.
    - LINUX: The host operating system is Linux.
    - NetWare: The host operating system is NetWare.
    - Solaris: The host operating system is Solaris and you are not using Solaris EFI labels.
    - Solaris_EFI: You are using Solaris EFI labels.

      Note that using any other LUN Multiprotocol Type with Solaris EFI labels might result in LUN misalignment. Refer to your Solaris Host Utilities documentation and release notes for more information.
    - VMware: You are using ESX Server and your LUNs will be configured with VMFS.

      If you configure the LUNs with raw disk mapping, use the guest operating system as the LUN Multiprotocol Type.
    - Windows: The host operating system is Windows 2000 Server, Windows XP, or Windows Server 2003 using the master boot record (MBR) partitioning method.
    - Windows_2008: The host operating system is Windows Server 2008; both MBR and GUID Partition Table (GPT) partitioning methods are supported.
    - Windows_GPT: You want to use the GPT partitioning method and your host is capable of using it. Windows Server 2003, Service Pack 1 and later, is capable of using the GPT partitioning method, and all 64-bit versions of Windows support it.
    - Image: The host operating system is not specified in the list, but it is listed as a supported operating system.

Published data

The following table indicates the output that is available from the **Create a LUN by Size** activity:

Name	**Type**
Backing snapshot	String
Block size	Integer
Comment	String
Is space reservation enabled	Boolean
Mapped	Boolean
Multiprotocol Type	Enumerated OS type
Online	Boolean
Path	String
Read only	Boolean
Serial number	String
Share state	String
Size	Decimal
UUID	String
Results message	String
Results status	Enumerated OS type

1. Operating system types and multiprotocol operating system types: The LUN Multiprotocol Type, or operating system type, specifies the operating system of the host that is accessing the LUN. It also determines the layout of data on the LUN, the geometry used to access that data, and the minimum and maximum size of the LUN.

- AIX: The host operating system is AIX.
- HPUX: The host operating system is HPUX.
- Hyper-V: You are using Windows Server 2008 Hyper-V and your LUNs contain virtual hard disks (VHDs). For raw LUNs, use the child operating system type as the LUN Multiprotocol Type.
- LINUX: The host operating system is Linux.
- NetWare: The host operating system is NetWare.
- Solaris: The host operating system is Solaris and you are not using Solaris EFI labels.
- Solaris_EFI: You are using Solaris EFI labels.

Note that using any other LUN Multiprotocol Type with Solaris EFI labels may result in LUN misalignment problems. Refer to your Solaris Host Utilities documentation and release notes for more information.

- VMware: You are using ESX Server and your LUNs will be configured with VMFS.

  If you configure the LUNs with raw disk mapping, use the guest operating system as the LUN Multiprotocol Type.
- Windows: The host operating system is Windows 2000 Server, Windows XP, or Windows Server 2003 using the master boot record (MBR) partitioning method.
- Windows_2008: The host operating system is Windows Server 2008; both MBR and GUID Partition Table (GPT) partitioning methods are supported.
- Windows_GPT: You want to use the GPT partitioning method and your host is capable of using it. Windows Server 2003, Service Pack 1 and later are capable of using the GPT partitioning method, and all 64-bit versions of Windows support it.
- Image: The host operating system is not specified in the list but it is listed as a supported operating system.

2. Operation statuses:

- 0 = Success
- 1 = Error
- 2 = Warning
- 3 = Idle
- 4 = ExecutionOn
- 5 = ExecutionOnWithErr
- 6 = ExecutionDone
- 7 = ExecutionDoneWithError
- 8 = ExecutionError
- 9 = CancelOn
- 10 = CancelErr
- 11 = Cancelled
- 12 = NotInitialized
- 13 = Initialized
- 14 = Unknown

- 15 = Information
- 16 = NotApplicable
- 17 = ErrorCritical
- 18 = ConflictRemoved
- 19 = ForceNotSpecified
- 20 = CleanupDone
- 21 = ExecutionDoneWithWarning

Create a LUN Clone activity

You use the **Create a LUN Clone** activity to create a clone of an existing LUN. You can use the clone, which is a complete copy of the LUN, for data backup.

User input

The following table indicates the information that you can enter in the properties tab for the **Create a LUN Clone** activity:

Property name	**Type**	**Required?**	**Default**
Controller name [1]	String	Required	
Parent path[2]	String	Required	
Clone path[2]	String	Required	
Parent snapshot	String	Required	
Unreserved	Boolean	Optional	False

1. Can be a host name or an IP address
2. `/vol/`*`volName/LUNName`*

Published data

The following table indicates the output that is available from the **Create a LUN Clone** activity:

Name	**Type**
Backing snapshot	String
Block size	Integer
Comment	String

Name	**Type**
Is space reservation enabled	Boolean
Mapped	Boolean
Multiprotocol type[1]	Enumerated OS Type
Online	Boolean
Path	String
Read only	Boolean
Serial number	String
Share state	String
Size	Decimal
UUID	String
Results message	String
Results status[2]	Enumerated Operation status

1. Operating system types and multiprotocol operating system types. The LUN Multiprotocol Type, or operating system type, specifies the operating system of the host that is accessing the LUN. It also determines the layout of data on the LUN, the geometry used to access that data, and the minimum and maximum size of the LUN.

   - AIX: The host operating system is AIX.
   - HPUX: The host operating system is HPUX.
   - Hyper-V: You are using Windows Server 2008 Hyper-V and your LUNs contain virtual hard disks (VHDs). For raw LUNs, use the child operating system type as the LUN Multiprotocol Type.
   - LINUX: The host operating system is Linux.
   - NetWare: The host operating system is NetWare.
   - Solaris: The host operating system is Solaris and you are not using Solaris EFI labels.
   - Solaris_EFI: You are using Solaris EFI labels.

     Note that using any other LUN Multiprotocol Type with Solaris EFI labels may result in LUN misalignment problems. Refer to your Solaris Host Utilities documentation and release notes for more information.
   - VMware: You are using ESX Server and your LUNs will be configured with VMFS.

If you configure the LUNs with raw disk mapping, use the guest operating system as the LUN Multiprotocol Type.

- Windows: The host operating system is Windows 2000 Server, Windows XP, or Windows Server 2003 using the master boot record (MBR) partitioning method.
- Windows_2008: The host operating system is Windows Server 2008; both MBR and GUID Partition Table (GPT) partitioning methods are supported.
- Windows_GPT: You want to use the GPT partitioning method and your host is capable of using it. Windows Server 2003, Service Pack 1 and later, is capable of using the GPT partitioning method, and all 64-bit versions of Windows support it.
- Image: The host operating system is not specified in the list but it is listed as a supported operating system.

2. Operational statuses:

- 0 = Success
- 1 = Error
- 2 = Warning
- 3 = Idle
- 4 = ExecutionOn
- 5 = ExecutionOnWithErr
- 6 = ExecutionDone
- 7 = ExecutionDoneWithError
- 8 = ExecutionError
- 9 = CancelOn
- 10 = CancelErr
- 11 = Cancelled
- 12 = NotInitialized
- 13 = Initialized
- 14 = Unknown
- 15 = Information
- 16 = NotApplicable
- 17 = ErrorCritical

- 18 = ConflictRemoved
- 19 = ForceNotSpecified
- 20 = CleanupDone
- 21 = ExecutionDoneWithWarning

Create a Volume Clone activity

You use the **Create a Volume Clone** activity to create a volume that is a clone (a writable copy) of a parent volume. Initially, the clone and its parent share the same storage; more storage space is consumed only as one volume or the other changes. By default, LUNs that are in the cloned volume are in an offline state.

Cloning applies exclusively to FlexVol volumes and requires the FlexClone license.

Properties

The following table indicates the information that you can enter in the properties tab for the **Create a Volume Clone** activity:

Property name	Type	Required?	Default
Controller name [1]	String	Required	
Parent volume	String	Required	
Clone volume	String	Required	
Parent snapshot [2]	String	Optional	
Space reserve	Enumerated (none, file, volume)	Optional	Volume

1. Can be a host name or an IP address
2. If a specific Snapshot name within the parent volume is provided, it is chosen as the parent copy. Otherwise, the storage system creates a new, distinctly named copy in the parent volume.

Published data

The following table indicates the output that is available from the **Create a Volume Clone** activity:

Name	Type
Name	String
Disk count	Integer

Name	**Type**
Dedupe enabled	Boolean
Size available	Decimal
Size used	Decimal
Size total	Decimal
State	String
Disk count	Integer
Type	String
Containing aggregate	String
Results message	String
Results status [1]	Enumerated Operation status

1. Operational statuses:
    - 0 = Success
    - 1 = Error
    - 2 = Warning
    - 3 = Idle
    - 4 = ExecutionOn
    - 5 = ExecutionOnWithErr
    - 6 = ExecutionDone
    - 7 = ExecutionDoneWithError
    - 8 = ExecutionError
    - 9 = CancelOn
    - 10 = CancelErr
    - 11 = Cancelled
    - 12 = NotInitialized
    - 13 = Initialized
    - 14 = Unknown

- 15 = Information
- 16 = NotApplicable
- 17 = ErrorCritical
- 18 = ConflictRemoved
- 19 = ForceNotSpecified
- 20 = CleanupDone
- 21 = ExecutionDoneWithWarning

Create a Volume activity

You use the **Create a volume** activity to create a new FlexVol volume with the name and characteristics that you specify. You can create up to 200 volumes on each storage system.

Properties

The following table indicates the information that you can enter in the properties tab for the **Create a volume** activity:

Property name	Type	Required?	Default
Controller name [1]	String	Required	
Name	String	Required	
Aggregate	String	Required	
Size	String	Required	
Unit type [2]	Enumerated	Required	GB
SnapLock	Boolean	Optional	False
Language code	String	Optional	
Space reserve [3]	Enumerated	Optional	
SnapLock type [4]	Enumerated	Optional	

1. Can be a host name or an IP address
2. KB, MB, GB, and TB
3. None, file, volume
4. Compliance, enterprise

Published data

The following table indicates the output that is available from the **Create a volume** activity:

Name	**Type**
Name	String
Dedupe enabled	Boolean
State	String
Size available	Decimal
Size used	Decimal
Size total	Decimal
Disk count	Integer
Type	String
Containing aggregate	String
Results message	String
Results status [1]	Enumerated Operation status

1. Operational statuses:

- 0 = Success
- 1 = Error
- 2 = Warning
- 3 = Idle
- 4 = ExecutionOn
- 5 = ExecutionOnWithErr
- 6 = ExecutionDone
- 7 = ExecutionDoneWithError
- 8 = ExecutionError
- 9 = CancelOn
- 10 = CancelErr
- 11 = Cancelled

- 12 = NotInitialized
- 13 = Initialized
- 14 = Unknown
- 15 = Information
- 16 = NotApplicable
- 17 = ErrorCritical
- 18 = ConflictRemoved
- 19 = ForceNotSpecified
- 20 = CleanupDone
- 21 = ExecutionDoneWithWarning

Create an Aggregate activity

You use the **Create an aggregate** activity to create and name a new aggregate. You can create up to 200 aggregates on each storage system. The new aggregate might not be immediately operational; you can use the **List Aggregate Info** activity to query its status.

Note: If the message `ECANT_USE_ALL_DISKS` is returned, the requested aggregate was created but without the number of disks that were specified in the activity properties.

Properties

The following table indicates the information that you can enter in the properties tab for the **Create an aggregate** activity:

Property name	**Type**	**Required?**	**Default**
Controller name [1]	String	Required	
Aggregate name	String	Required	
Disk count [2]	Integer	Required	
Disks [3]	String[ ]	Required	
Mirror disks [4]	String[ ]	Optional	
Disk size [5]	Integer	Optional	
Unit type [6]	Enumerated	Optional	GB
Disk type [7]	Enumerated disk type	Optional	

Property name	**Type**	**Required?**	**Default**
Force spare pool	Boolean	Optional	False
Mirrored [8]	Boolean	Optional	False
SnapLock [9]	Boolean	Optional	False
Language code [10]	String	Optional	
RAID size	Integer	Optional	
RAID type [11]	Enumerated	Optional	
RPM	Integer		
SnapLock type [12]	Enumerated	Optional	
Type [13]	Enumerated	Optional	Aggregate
Use 64 bit [14]	Boolean	Optional	64 bit

1. Can be a host name or an IP address

2. You must specify the disk count or disk property

3. You must specify the disk count or disk property. If the aggregate is mirrored and a specific list of disks is supplied, another list (“mirror disks”) must also be supplied with the same number of disks.

4. If the aggregate is mirrored and a specific list of disks is supplied, the mirror disks property must also be supplied with the same number of disks.

5. If disk size is not specified, the plug-in selects the next disk that fits the other parameters, such as disk count, disk type, and so on.

6. KB, MB, GB, and TB

7. If disk type is not specified, the plug-in selects the next disk that fits the other parameters, such as disk count, disk type, and so on.

8. Requires a mirror license: syncmirror_local

9. - If SnapLockType is set, you must create the type specified in SnapLockType.
   - If SnapLockType is not set and a SnapLock Enterprise license is installed, you must create a SnapLock Enterprise aggregate.
   - If SnapLockType is not set and a SnapLock Enterprise license has not been installed, you must create a SnapLock Compliance aggregate.

   Requires a compliance or enterprise license

10. Must append .UTF-8 for UTF-8 as the NFS character set
11. raid4, raid_dp
12. Compliance, enterprise
13. Aggregate, traditional
14. Requires Data ONTAP 8.0 or later

Published data

The following table indicates the output that is available from the **Create an aggregate** activity:

Name	Type
Name	String
Disk count	Integer
Mirror status	String
Size available	Decimal
Size used	Decimal
Size total	Decimal
State	String
Volume count	Integer
Volumes	String
Results message	String
Results status [1]	Enumerated Operation status

1. Operational statuses:
   - 0 = Success
   - 1 = Error
   - 2 = Warning
   - 3 = Idle
   - 4 = ExecutionOn
   - 5 = ExecutionOnWithErr
   - 6 = ExecutionDone

- 7 = ExecutionDoneWithError
- 8 = ExecutionError
- 9 = CancelOn
- 10 = CancelErr
- 11 = Cancelled
- 12 = NotInitialized
- 13 = Initialized
- 14 = Unknown
- 15 = Information
- 16 = NotApplicable
- 17 = ErrorCritical
- 18 = ConflictRemoved
- 19 = ForceNotSpecified
- 20 = CleanupDone
- 21 = ExecutionDoneWithWarning

Delete a LUN activity

You use the **Delete a LUN** activity to delete a specified LUN when you no longer need it. This operation fails if the LUN is currently mapped and is online. However, if you must delete a mapped and online LUN, you can use the `-Force` option.

Properties

The following table indicates the information that you can enter in the properties tab for the **Delete a LUN** activity:

Property name	Type	Required?	Default
Controller name [1]	String	Required	
LUN path [2]	String	Required	
Force [3]	Boolean	Optional	False

1. Can be a host name or an IP address

2. `/vol/`*`volName/LUNName`*

3. Deletes the volume, even if it is online

Published data

The following table indicates the output you might get from the **Delete a LUN** activity:

Name	**Type**
Results message	String
Results status [1]	Enumerated Operation status

1. Operational statuses:
   - 0 = Success
   - 1 = Error
   - 2 = Warning
   - 3 = Idle
   - 4 = ExecutionOn
   - 5 = ExecutionOnWithErr
   - 6 = ExecutionDone
   - 7 = ExecutionDoneWithError
   - 8 = ExecutionError
   - 9 = CancelOn
   - 10 = CancelErr
   - 11 = Cancelled
   - 12 = NotInitialized
   - 13 = Initialized
   - 14 = Unknown
   - 15 = Information
   - 16 = NotApplicable
   - 17 = ErrorCritical

- 18 = ConflictRemoved
- 19 = ForceNotSpecified
- 20 = CleanupDone
- 21 = ExecutionDoneWithWarning

Delete a Volume activity

You use the **Delete a Volume** activity to delete a specified volume or plex when you do not need it anymore.

The following list provides guidelines for using the **Delete a Volume** activity:

- Only offline volumes and plexes can be deleted.
  To delete an online volume, you must specify the `Force` option to first take the volume offline.
- If you specify a FlexVol volume, all of the blocks in that volume are dissociated and returned to the aggregate that contained them; no other FlexVol volumes in the same aggregate are affected.
- If you specify a traditional volume, all of the plexes are deleted, and the disks in the volume are returned to the appropriate spare pools.
- If you specify a plex, it must be for a mirrored aggregate, which results in the aggregate being unmirrored.

Properties

The following table indicates the information that can be included in the properties tab for the **Delete a Volume** activity:

Property name	**Type**	**Required?**	**Default**
Controller name [1]	String	Required	
Volume name	String	Required	
Force [2]	Boolean	Optional	False

1. Can be a host name or an IP address
2. Deletes the volume, even if it is online

Published data

The following table indicates the output that is available from the **Delete a Volume** activity:

Name	**Type**
Results message	String
Results status [1]	Enumerated Operation status

1. Operational statuses:

- 0 = Success
- 1 = Error
- 2 = Warning
- 3 = Idle
- 4 = ExecutionOn
- 5 = ExecutionOnWithErr
- 6 = ExecutionDone
- 7 = ExecutionDoneWithError
- 8 = ExecutionError
- 9 = CancelOn
- 10 = CancelErr
- 11 = Cancelled
- 12 = NotInitialized
- 13 = Initialized
- 14 = Unknown
- 15 = Information
- 16 = NotApplicable
- 17 = ErrorCritical
- 18 = ConflictRemoved
- 19 = ForceNotSpecified
- 20 = CleanupDone
- 21 = ExecutionDoneWithWarning

Delete an Aggregate activity

You use the **Delete an Aggregate** activity to destroy a specified aggregate when you do not need it anymore.

The following list provides guidelines for using the **Delete an Aggregate** activity:

- You can delete only offline aggregates and plexes.
  You can use the `Force` option to take the volume offline.
- If you specify an aggregate, all plexes in the aggregate are deleted.
  If the aggregate is embedded in a traditional volume, the entire traditional volume is deleted. The operation is rejected if there FlexVol volumes in the aggregate.
- If you specify a plex, only that plex is deleted, leaving an unmirrored aggregate containing the remaining plex.
  The disks in the deleted object become spare disks.

Properties

The following table indicates the information that you can enter in the properties tab for the **Delete an Aggregate** activity:

Property name	**Type**	**Required?**	**Default**
Controller name [1]	String	Required	
Aggregate or plex name [2]	String	Optional	
Force [3]	Boolean	Optional	False

1. Can be a host name or an IP address
2. **`[/vol/]<aggrname>[<plexinfo>]`**
3. Deletes the aggregate, even if online

Published data

The following table indicates the output that is available from the **Delete an Aggregate** activity:

Name	**Type**
Results message	String
Results status [1]	Enumerated Operation status

1. Operational statuses:

    - 0 = Success
    - 1 = Error
    - 2 = Warning
    - 3 = Idle
    - 4 = ExecutionOn
    - 5 = ExecutionOnWithErr
    - 6 = ExecutionDone
    - 7 = ExecutionDoneWithError
    - 8 = ExecutionError
    - 9 = CancelOn
    - 10 = CancelErr
    - 11 = Cancelled
    - 12 = NotInitialized
    - 13 = Initialized
    - 14 = Unknown
    - 15 = Information
    - 16 = NotApplicable
    - 17 = ErrorCritical
    - 18 = ConflictRemoved
    - 19 = ForceNotSpecified
    - 20 = CleanupDone
    - 21 = ExecutionDoneWithWarning

List Aggregate Info activity

You use the **List Aggregate Info** activity to retrieve information about one or more aggregates from a Data ONTAP controller. You can use wildcards in the aggregate name to list the aggregates.

Properties

The following table indicates the information that you can enter in the properties tab for the **List Aggregate Info** activity:

Property name	**Type**	**Required?**	**Default**
Controller name [1]	String	Required	
Aggregate names	String[ ]	Optional	Defaults to all aggregates on the controller

1. Can be a host name or an IP address

Published data

The following table indicates the output that is available from the **List Aggregate Info** activity:

Name	**Type**
Name	String
Disk count	Integer
Mirror status	String
Size available	Decimal
Size used	Decimal
Size total	Decimal
State	String
Volume count	Integer
Volumes	String
Results message	String
Results status [1]	Enumerated Operation status

1. Operational statuses:

- 0 = Success
- 1 = Error
- 2 = Warning
- 3 = Idle
- 4 = ExecutionOn
- 5 = ExecutionOnWithErr
- 6 = ExecutionDone
- 7 = ExecutionDoneWithError
- 8 = ExecutionError
- 9 = CancelOn
- 10 = CancelErr
- 11 = Cancelled
- 12 = NotInitialized
- 13 = Initialized
- 14 = Unknown
- 15 = Information
- 16 = NotApplicable
- 17 = ErrorCritical
- 18 = ConflictRemoved
- 19 = ForceNotSpecified
- 20 = CleanupDone
- 21 = ExecutionDoneWithWarning

List LUN Info activity

You use the **List LUN info** activity to capture LUN information (such as size, state, comments, serial number, LUN mapping, and so on) about the specified LUN or about all LUNs.

Properties

The following table indicates the information that you can enter in the properties tab for the **List LUN info** activity:

Property name	Type	Required?	Default
Controller name [1]	String	Required	
LUN path [2]	String	Optional	

1. Can be a host name or an IP address
2. `/vol/`*`volName/LUNName`*; wildcards permitted

Published data

The following table indicates the output that is available from the **List LUN info** activity:

Name	Type
Backing snapshot	String
Block size	Integer
Comment	String
Is space reservation enabled	Boolean
Mapped	Boolean
Multiprotocol Type [1]	Enumerated OS type
Online	Boolean
Path	String
Read only	String
Results message	String
Results status [2]	Enumerated Operation status
Serial number	String

Name	Type
Share state	String
Size	Decimal
Size used	Decimal
UUID	String

1. Operating system types and multiprotocol operating system types. The LUN Multiprotocol Type, or operating system type, specifies the operating system of the host that is accessing the LUN. It also determines the layout of data on the LUN, the geometry used to access that data, and the minimum and maximum size of the LUN.

- AIX: The host operating system is AIX.
- HPUX: The host operating system is HPUX.
- Hyper-V: You are using Windows Server 2008 Hyper-V and your LUNs contain virtual hard disks (VHDs). For raw LUNs, use the child operating system type as the LUN Multiprotocol Type.
- LINUX: The host operating system is Linux.
- NetWare: The host operating system is NetWare.
- Solaris: The host operating system is Solaris and you are not using Solaris EFI labels.
- Solaris_EFI: You are using Solaris EFI labels.

  Note that using any other LUN Multiprotocol Type with Solaris EFI labels may result in LUN misalignment problems. Refer to your Solaris Host Utilities documentation and release notes for more information.
- VMware: You are using ESX Server and your LUNs will be configured with VMFS.

  If you configure the LUNs with raw disk mapping, use the guest operating system as the LUN Multiprotocol Type.
- Windows: The host operating system is Windows 2000 Server, Windows XP, or Windows Server 2003 using the master boot record (MBR) partitioning method.
- Windows_2008: The host operating system is Windows Server 2008; both MBR and GUID Partition Table (GPT) partitioning methods are supported.
- Windows_GPT: You want to use the GPT partitioning method and your host is capable of using it. Windows Server 2003, Service Pack 1 and later, is capable of using the GPT partitioning method, and all 64-bit versions of Windows support it.
- Image: The host operating system is not specified in the list but it is listed as a supported operating system.

2. Operational statuses:

    - 0 = Success
    - 1 = Error
    - 2 = Warning
    - 3 = Idle
    - 4 = ExecutionOn
    - 5 = ExecutionOnWithErr
    - 6 = ExecutionDone
    - 7 = ExecutionDoneWithError
    - 8 = ExecutionError
    - 9 = CancelOn
    - 10 = CancelErr
    - 11 = Cancelled
    - 12 = NotInitialized
    - 13 = Initialized
    - 14 = Unknown
    - 15 = Information
    - 16 = NotApplicable
    - 17 = ErrorCritical
    - 18 = ConflictRemoved
    - 19 = ForceNotSpecified
    - 20 = CleanupDone
    - 21 = ExecutionDoneWithWarning

List Volume Info activity

You use the **List Volume Info** activity to get the volume status. All RAID-related status items that are reported for a volume actually describe the state of the aggregate that contains the volume. The output is a list of volumes in a particular aggregate, information that can be used to determine whether a volume exists and used as input to additional operations.

Properties

The following table indicates the information that you can enter in the properties tab for the **List Volume Info** activity:

Property name	Type	Required?
Controller name [1]	String	Required
Volume names [2]	String[]	Optional
Aggregate[2]	String	Optional

1. Can be a host name or an IP address
2. Wildcards permitted

Published data

The following table indicates the output that is available from the **List Volume Info** activity:

Name	Type
Name	String
Containing aggregate	String
Dedupe enabled	Boolean
Disk count	Integer
Size available	Decimal
Size used	Decimal
Size total	Decimal
Type	String
Results message	String
Results status [1]	Enumerated Operation status

1. Operational statuses:
   - 0 = Success
   - 1 = Error

- 2 = Warning
- 3 = Idle
- 4 = ExecutionOn
- 5 = ExecutionOnWithErr
- 6 = ExecutionDone
- 7 = ExecutionDoneWithError
- 8 = ExecutionError
- 9 = CancelOn
- 10 = CancelErr
- 11 = Cancelled
- 12 = NotInitialized
- 13 = Initialized
- 14 = Unknown
- 15 = Information
- 16 = NotApplicable
- 17 = ErrorCritical
- 18 = ConflictRemoved
- 19 = ForceNotSpecified
- 20 = CleanupDone
- 21 = ExecutionDoneWithWarning

Storage system credential cmdlets and activities

You can use the storage systems credential cmdlets or activities to add credentials to the database so that you can access controllers. Every administration user on the host where the file is located can invoke provisioning and cloning and storage-side operations with the credentials stored in the file.

What the storage system credential cmdlets are

The storage system credential cmdlets enable you to add, list, and remove the cached credentials to access your storage system. The storage cache cmdlets `Add-OCStorageSystem`, `Get-OCStorageSystem`, and `Remove-OCStorageSystem` support a remote `-Server` switch parameter that supports PowerShell cmdlets.

Credentials to access a specific storage system are encrypted and saved in a database. All administration users on the host where the file is located can use the credentials to invoke provisioning and cloning cmdlets and storage-side operations.

Add-OCStorageSystem cmdlet

The `Add-OCStorageSystem` cmdlet adds user credentials to the plug-in database so that you can use other OnCommand Plug-in for Microsoft cmdlets without entering additional password information. `Add-OCStorageSystem` supports clustered Data ONTAP.

Syntax

```
Add-OCStorageSystem
[<-StorageSystem, system> [<String>] Storage_system_name_or_IP_address]
[<-User, usr> [<String>] User_name]
[<-Password, pwd> [<SecureString>] User_password]
[<-Protocol, pro> [<String>] HTTPS | HTTP
[<CommonParameters>]
```

Description

The `Add-OCStorageSystem` cmdlet adds or modifies the specified storage system in the OnCommand Plug-in for Microsoft database.

The credentials persist in the OnCommand Plug-in for Microsoft database with encryption.

For more information, see the Microsoft TechNet web site.

Parameters

`-StorageSystem` | `-system` *`Storage_system_name_or_IP_address`*

Specifies the storage system to which you want to add credentials. If you are cloning on clustered Data ONTAP, you can add an SVM or cluster.

Required?

True

Position

0

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-User` | `-usr` ***`User_name`***

Specifies a valid user name on the storage system.

Required?

True

Position

1

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-Password` ***`User_password`***

Specifies the password associated with the storage system user name. The user password must be a secure string.

Required?

True

Position

2

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-Protocol` `HTTPS` | `HTTP`

Specifies the type of protocol that you want to use. The default is HTTP.

Required?

False

Position

3

Default value

HTTPS

Accept pipeline input?

False

Accept wildcard characters?

False

Example: Adding storage system credentials

This cmdlet adds the storage systems *`myFiler`*, the user *`myUserName`*, and the password *`myPassword`* to the OnCommand Plug-in for Microsoft database:

```
Add-OCStorageSystem -StorageSystem myFiler -user myUserName -Password
myPassword -Protocol HTTPS
```

Get-OCStorageSystem

The `Get-OCStorageSystem` cmdlet lists all of the credentials that are stored in the OCPM database or all the credentials for specified storage systems. Every administration user of the host on which the file is located can invoke provisioning and cloning cmdlets and storage-side operations using the credentials stored in the OCPM database.

Syntax

```
Get-OCStorageSystem
[<-StorageSystem, system> [<String>]Storage_system_name_or_IP_address][<-
HostType, ht> [<String>]]
```

Description

This cmdlet provides either specific protocol information, storage system names, and user names that are stored in the OCPM database, or it provides all of the credentials for a specified storage system.

Parameters

`-StorageSystem` | `-system` *`Storage_system_name_or_IP_address`* (Optional)

Displays all credentials currently stored in the OCPM database or credentials for a specified storage system.

Required?

False

Position

0

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-HostType` | `ht` (Optional)

Specifies the host type of the storage system:

- **Controller**: Valid for 7-Mode
- **vFiler**: Valid for 7-Mode
- **Cluster**: Valid for clustered Data ONTAP
- **SVM**: Valid for clustered Data ONTAP

Required?

False

Position

1

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

Example: Getting storage system credentials

The following cmdlet displays the credentials for *`StorageSystemA`*:

```
Get-OCStorageSystem-StorageSystem StorageSystemA
The input parameters validation successful.
User credentials are set for the following storage system.
StorageSystem: StorageSystemA
User: ProvisionUser
Protocol: http
```

Remove-OCStorageSystem cmdlet

The `Remove-OCStorageSystem` cmdlet removes the credentials for a specified storage system from the plug-in database. The proper use of the `Remove-OCStorageSystem` cmdlet can increase security and prevent unauthorized users from accessing the storage system.

Even if you have multiple IP addresses configured on the storage system, the `Remove-OCStorageSystem` cmdlet removes the specified storage system. You can also remove the storage system using the host name, in which case the cmdlet removes all the entries that the IP addresses associate with the host name.

Syntax

```
Remove-OCStorageSystem
[<-StorageSystem, system> <String> Storage_system_name_or_IP_address]
```

Parameters

`-StorageSystem` | `-system` *`Storage_system_name_or_IP_address`* (Optional)

Specifies the storage system credentials that you want to remove.

Required?

False

Position

0

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

Example: Removing storage system credentials

The following cmdlet removes the credentials for *`StorageSystemA`* from the plug-in database:

```
Remove-OCStorageSystem-StorageSystem StorageSystemA
```

```
PS C:\Users\administrator.SDDEV> Remove-OCStorageSystem
-StorageSystem tt -verbose
VERBOSE: Starting Remove-OCStorageSystem
VERBOSE: Processing Remove-OCStorageSystem..
VERBOSE: Performing operation "Removing StorageSystem..." on Target
tt.

Operationstatus    : Success
Type               :
HelpLink           :
Source             :
Resolution         :
RecoverableError   : True
Description        :
StackTrace         :
Detail             :
Message            : Remove-OCStorageSystem: Successfully removed
the <tt> StorageSystem.
ErrorCode          : 0
_message           : Remove-OCStorageSystem: Successfully removed
the <tt> StorageSystem.
_errorCode         : 0
RecoverableError   : True
ShowError          : False
```

What storage system credential activities are

The storage system credential activities enable you to add, remove, and list stored storage system credentials. Removing storage system credentials makes it necessary to manually input passwords each time the specified storage system is accessed.

Add Controller Credentials activity

You use the **Add Controller Credentials** activity to add credentials to access a controller. Using credentials enforces the security of your data.

Properties

The following table indicates the information that you can enter in the properties tab for the **Add Controller Credentials** activity:

Property name	Type	Required?	Default
Controller name [1]	String	Required	None
User name	String	Required	None

Property name	Type	Required?	Default
Password	String	Required	None
Protocol [2]	Choice	Required	HTTPS

1. Indicates the name or the IP address of the controller.
2. Indicates the protocol to use for the credential.

Published data

The following table indicates the output that is available from the **Add Controller Credentials** activity:

Name	Type	Format
Controller name	String	

Remove Controller Credentials activity

You use the **Remove Controller Credentials** activity to designate who can—and cannot— access a controller.

Even if you have multiple IP addresses configured on the storage system, the **Remove Controller Credentials** activity removes the specified storage system. You can also remove the storage system using the host name, in which case the cmdlet removes all the entries that the IP addresses associate with the host name.

Properties

The following table indicates the information that you can enter in the properties tab for the **Remove Controller Credentials** activity:

Property name	Type	Required?	Default
Controller name [1]	String	Required	None

1. Indicates the name or the IP address of the controller

Published data

The following table indicates the output that is available from the **Remove Controller Credentials** activity:

Name	Type
Controller name	String

List Controller Credentials activity

You use the **List Controller Credentials** activity to list all of the storage system credentials that are stored in the OCPM database. Every administrator on the host where the database is located can invoke provisioning, cloning, and disaster recovery activities and storage-side operations using the credentials stored in the OCPM database.

Properties

The following table lists the information that you can enter in the properties tab for the **List Controller Credentials** activity:

Property name	**Type**	**Required?**	**Default**
Controller name [1]	String	Optional	None

1. Indicates the name or the IP address of the controller

Published data

The following table indicates the information that the plug-in can return from the **List Controller Credentials** activity:

Name	**Type**
Controller name	String
Protocol	String
User name	String

Provisioning cmdlets and activities

You can use the PowerShell provisioning cmdlets with Microsoft System Center Virtual Machine Manager (SCVMM) applications to perform a variety of provisioning tasks to improve performance and space efficiency on your Hyper-V virtual machines. Some of these tasks include configuring objects for the web service, creating, connecting, resizing, and disconnecting storage.

Provisioning storage

By using provisioning cmdlets with Microsoft System Center Virtual Machine Manager (SCVMM) applications, you can use your existing storage resources to perform faster and more space-efficient provisioning of Hyper-V virtual machines. Provisioning in OnCommand Plug-in for Microsoft is only supported for Data ONTAP 7-Mode.

Related concepts

Windows PowerShell execution policies and administration profiles on page 6

Connect-OCStorage cmdlet

The `Connect-OCStorage` cmdlet connects an existing LUN on a storage system to a specified host system or server to increase your accessibility to the data. It can also connect storage to multiple nodes of a cluster. Only LUNs that were previously mounted on the host can be connected with the `Connect-OCStorage` cmdlet.

Syntax

```
Connect-OCStorage
[<-Server, s> <string> Server_name]
[<-StoragePath, p, path> <string> Storage_path_name]
[<-MountPoint, mount, mp, d> <string> LUN_mount_point]
[<-InitiatorName, initiator, i> Initiator_name]
[<-ResourceName, rn> <string> resource_name]
[<-ResourceGroupName, rg> <string> resource_group_name]
[<-ClusterSharedVolume, csv>]
[<CommonParameters>]
```

Parameters

`-Server` | `-s` *`Server_name`*

Specifies the name of the host or Microsoft failover cluster to which you want to provision storage.

When the server name is the name of a cluster, the LUN is connected to all nodes of the cluster and the disk is configured as shared in a Windows failover cluster. The default is the name of the local machine.

Connected storage appears in the Available Storage cluster group.

If `-csv` is specified, the disk is changed to a clustered shared volume and it appears in the **`Clustered Shared Volumes`** group.

Required?

False

Position

Named

Default value

The local machine

Accept pipeline input?

False

Accept wildcard characters?

False

`-StoragePath |-path | -p` ***`Storage_path_name`***

Specifies the full LUN path, including the storage system name, volume name, and LUN name.

Required?

True

Position

Named

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-MountPoint |-mount | -mp | -d` ***`LUN_mount_point`***

Specifies the LUN mount point. The default is the volume GUID. You can use either a drive letter or an NTFS folder mount path. If you use a drive letter, that drive letter must not already be in use.

Required?

True

Position

Named

Default value

A volume GUID

Accept pipeline input?

False

Accept wildcard characters?

False

`-InitiatorName` | `-i` *`Initiator_name`*

Specifies the name of the initiator or initiators to which the LUN is mapped. If you do not specify an initiator and there is a Fibre Channel connection that is enabled between the host and the storage controller, the Fibre Channel initiator name is used. If a Fibre Channel initiator name is not used, either the first available iSCSI initiator with an established session is used or a new iSCSI session is created. You should consider the following protocol restrictions:

- You must zone the initiator port properly with the target port from the storage system.
- If you use iSCSI, the initiator name is the IQN name.
- If you use Fibre Channel, the initiator name is the worldwide port name (WWPN).
  The plug-in picks one Fibre Channel initiator from each node of the cluster.
- You must enable MPIO to specify more than one initiator from the same host.
  If MPIO is not installed and you have multiple FCP initiators from the same host, you should not specify the initiator parameters explicitly. You should specify the cluster name as the server parameter for the plug-in to handle it correctly.
- If you specify invalid or disabled initiators, the storage provisioning operation fails with an error message that indicates which initiator is invalid.

Required?

False

Position

Named

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-ResourceName` | **`-rn`** ***`Resource_name`***

Specifies the Windows failover cluster resource name. If `ResourceName` is not specified, *`Cluster Disk n`* is used, where *`n`* is an integer.

Required?

False

Position

Named

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-ResourceGroupName` | **`-rg`** ***`Resource_group_name`***

Specifies the name of the resource group, which is a group that contains several Windows failover cluster resources that all fail over together. If a resource group with the specified name does not exist, a new group is created with the specified name. If a resource group name is not specified, *`Available Storage`* is used.

Required?

False

Position

Named

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-ClusterSharedVolume` | **`-csv`**

Specifies that a shared disk be added to the Cluster Shared Volume. When this parameter is specified, the `Server` parameter must be the name of a Windows failover cluster and the mount point must not be specified.

Required?

False

Position

Named

Default value

False

Accept pipeline input?

False

Accept wildcard characters?

False

Example: Connecting storage

The following cmdlet uses the path `/vol/vol2/lun2` on `storagesystemA` to connect `server1` to the LUN. It then uses the first available iSCSI session to mount the disk on mount point `M`.

```
Connect-OCStorage -Server server1 -StoragePath storagesystemA:/vol/
vol1/lun1 -Mountpoint M:\
```

Disconnect-OCStorage cmdlet

The `Disconnect-OCStorage` cmdlet disconnects a LUN from the specified host system or server. The LUN remains on the storage system. You can also disconnect storage from multiple nodes of the cluster and reconnect them as needed.

Syntax

```
Disconnect-OCStorage
[<-Server, s> [<string>] Server_name
[<-StoragePath, path, p> [<string>] Storage_path_name
[<-MountPoint, mount, mp, d> [<string>] LUN_mountpoint
[<-Force, f>] [<SwitchParameter>]
[<CommonParameters>]]
```

Parameters

`-Server | -s` ***`Server_name`***

Specifies the host server name or Microsoft failover cluster name of the host server from which to disconnect storage. The default is the local machine.

In the syntax for this cmdlet, when a disk is connected as a CSV or as a shared disk to a cluster, the `Server` parameter must be a cluster name, in which case the cluster resource that is dependent on the connected LUN is removed first and then the LUN is disconnected from all the nodes of the cluster.

Required?

False

Position

Named

Default value

The local machine

Accept pipeline input?

False

Accept wildcard characters?

False

`-StoragePath |-path | -p` ***`Storage_path_name`***

Specifies the full path of the LUN that you want to disconnect, including the LUN name, volume name, and storage system name. If you do not specify either the storage path or the mount point, the cmdlet fails.

Required?

False

Position

Named

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-MountPoint |-mount | -mp | -d` ***`LUN_mount_point`***

Specifies the LUN mount point. If you do not specify either the storage path or the mount point, the cmdlet fails.

Required?

False

Position

Named

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-Force` | `-f`

Specifies whether to forcefully disconnect the LUN when the file system in the LUN is in use, has virtual machines, or has open files. If you do not use this parameter, the file system cannot be unmounted due to the open file, and the LUN is not disconnected. The default value is **`False`**.

The `Force` parameter is always required for clustered LUNs or when the `Server` parameter is a remote server name.

Required?

False

Position

Named

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

Example: Disconnecting storage

The following cmdlet unmounts the disk on mount point *`M`* and disconnects *`server1`* from the LUN on *`storagesystemA`* using the path *`/vol/vol1/lun1`*:

```
Disconnect-OCStorage -Server server1 -StoragePath storagesystemA:/vol/
vol1/lun1 -Mountpoint M:\
```

Get-OCStorage cmdlet

The `Get-OCStorage` cmdlet generates a report with a list of the LUN paths that are mounted on a specified host system or server. You can use this cmdlet to get storage information from all nodes of the cluster.

Syntax

```
Get-OCStorage
[[-{Server, s}] <String> Server_name]
[[-{Mountpoint, mount, mp, d}] <String> Mount_point]
[[-{StoragePath, path, p}] <String> Storage_path]
```

```
[[-{Summary, smry}] [<SwitchParameter>]]
[[<CommonParameters>]]
```

Parameters

`-Server` | **`-s`** ***`Server_name`***

Specifies the name of the host server from which to discover and list storage. The default is the local machine.

Required?

False

Position

0

Default value

The local machine

Accept pipeline input?

False

Accept wildcard characters?

False

`-MountPoint` | **`-mount | -mp | -d`** ***`LUN_mount_point`***

Specifies the mount point for which to show the storage information.

Required?

False

Position

1

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-StoragePath` | **`-p`*****`Storage_path`***

Specifies the full LUN path, including the LUN name, volume name, and storage system name.

Required?

True

Position

2

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-Summary` | `-smry`

Generates a report with a list of the LUN paths, sorted by mount point.

Required?

False

Position

3

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

Example: Getting information about storage

The following cmdlet lists information about the LUNs that are mounted on *`server1`*:

```
Get-OCStorage -Server server1
```

New-OCStorage cmdlet

The `New-OCStorage` cmdlet provisions a new LUN from a storage system to a specified host system or server to expand your storage. You can also map storage to multiple nodes of a cluster to improve data access.

The LUN is created with the `SpaceReservation` option disabled. When the `SpaceReservation` option is enabled for one or more files or LUNs, Data ONTAP reserves enough space in the volume so that write operations to those files or LUNs do not fail due to a lack of disk space.

You can use either Fibre Channel or iSCSI protocol to connect the LUN and map it to multiple nodes of a cluster.

Syntax

```
New-OCStorage
[[<-Server, s>] <String> Server_name]
[[<-StoragePath, path, sp, s>] <String> Storage_path_name]
[[<-Size, sz, z>] <SwitchParameter> KB | MB | GB | TB]
[[<-MountPoint, path, mp, d>] <String> LUN_mountpoint]
[[<-InitiatorName, initiator, i>] <String> Initiator_name]
[[<-ResourceName, rn>] <String> Resource_name]
[[<-ResourceGroup, rg>] <String> Resource_group_name]
[[<-ClusterSharedVolume, csv>]
[[<-{SpaceReservation, sr}]]
[[<CommonParameters>]]
```

Parameters

-Server | -s *Server_name*

Specifies the name of the host or Microsoft failover cluster from which to connect storage.

When a cluster name is specified, the LUN is connected to all nodes of the cluster and the disk is configured as “shared”. The default is the local machine.

Connects storage displays in the Available Storage cluster group.

If the `ClusteredSharedVolume` parameter is specified, the disk is made a clustered shared volume and displays in the Clustered Shared Volumes group.

Required?

True

Position

Named

Default value

The local machine

Accept pipeline input?

False

Accept wildcard characters?

False

-StoragePath | -path | -p *Storage_path_name*

Specifies the full LUN path, including the LUN name, volume name, and storage system name.

Required?

True

Position

Named

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-Size |-z KB|MB|GB|TB`

Specifies the LUN size to provision.

Required?

True

Position

Named

Default value

GB

Accept pipeline input?

False

Accept wildcard characters?

False

`-MountPoint |-mount | -mp | -d` ***`LUN_mountpoint`***

Specifies the LUN mount point. The default is the volume GUID. You can use either a drive letter alone or a drive letter and directory as a mount point. If you use a drive letter and directory, the directory and the drive that are identified by that letter must already exist. If you use a drive letter alone, that drive letter must not already be in use.

Required?

False

Position

Named

Default value

The volume GUID

Accept pipeline input?

False

Accept wildcard characters?

False

`-InitiatorName | -i` ***`Initiator_name`***

Specifies the name of the initiator or initiators to which the LUN is mapped. If you do not specify an initiator and there is a Fibre Channel connection that is enabled between the host and the storage controller, the Fibre Channel initiator name is used. If a Fibre Channel initiator name is not used, either the first available iSCSI initiator with an established session is used or a new iSCSI session is created.

If you change the default iSCSI initiator name to one that does not contain the server host name, an existing iSCSI session must exist between the iSCSI host and the target controller for the plug-in to find the initiator based on the existing iSCSI sessions.

You should consider the following protocol restrictions:

- You must zone the initiator port properly with the target port from the storage system.
- If you use iSCSI, the initiator name must be the IQN name.
- If you use Fibre Channel, the initiator name must be the worldwide port name (WWPN).
  The plug-in picks one Fibre Channel initiator from each node of the cluster.
- You must enable MPIO to specify more than one initiator from the same server.
  If MPIO is not installed and you have multiple FCP initiators from the same host, you should not specify the initiator parameters explicitly. You should specify the cluster name as the server parameter for the plug-in to handle it correctly.
  If MPIO is not enabled, the plug-in picks one Fibre Channel initiator from each node of the cluster; otherwise, the plug-in puts all available Fibre Channel initiators from each node into an initiator group.
  Therefore, if MPIO is enabled, each cluster node has an initiator group that contains all its enabled and available Fibre Channel initiators.
- The storage provisioning operation fails with an error if you take any of the following actions:
  - Specify an initiator that does not belong to any of the cluster nodes.
  - Do not specify an initiator for a cluster node when you connect to a shared disk or CSV and MPIO is enabled.
  - Specify invalid or disabled initiators (the error message indicates which initiator is invalid).

Required?

False

Position

Named

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-ResourceName` | `-rn` *`Resource_name`*

Specifies the Windows failover cluster resource name. If the `-ResourceName` parameter is not specified, either a volume GUID or a specified drive letter is used for naming the resource.

Required?

False

Position

Named

Default value

The cluster disk

Accept pipeline input?

False

Accept wildcard characters?

False

`-ResourceGroup` | `-rg` *`Resource_group_name`*

Specifies the name of the resource group, which is a group that contains several Windows failover cluster resources that all fail over together. The `-ResourceGroupName` parameter is determined by the following factors:

- If a resource group with the specified name exists, a new share disk is added to that resource group.
- If a resource group with the specified name does not exist, a new group is created with the specified name.
- If the `-ResourceGroupName` parameter is not specified, a shared disk is added to available storage.

Required?

False

Position

Named

Default value

Available storage

Accept pipeline input?

False

Accept wildcard characters?

False

`-ClusterSharedVolume` | `-csv`

Specifies that a shared disk be added to the Cluster Shared Volume. When this parameter is specified, the `-server` parameter must be the name of a Windows failover cluster.

Required?

False

Position

Named

Default value

False

Accept pipeline input?

False

Accept wildcard characters?

False

`- SpaceReservation` | `-sr`

If specified, the LUN is created with space being reserved. By default, the LUN is created with Space Reservation disabled unless the `-SpaceReservation` parameter is specified.

Required?

False

Position

Named

Default value

False

Accept pipeline input?

False

Accept wildcard characters?

False

Example: Creating new storage

The following cmdlet creates a 10-GB LUN on `storagesystemA` using the path `/vol/vol1/lun1`. It then mounts the disk on mount point `M` and uses the first available iSCSI session to connect `server1` to the LUN.

```
New-OCStorage -Server server1 -StoragePath storagesystemA:/vol/vol1/
lun1 -Size 10GB -Mountpoint M:\
```

Remove-OCStorage cmdlet

The `Remove-OCStorage` cmdlet disconnects a LUN from the host system or server and deletes the LUN from the storage system. You can use this cmdlet to delete all user data from multiple nodes of a cluster.

Syntax

```
Remove-OCStorage
[[<-Server, s>] <String> Server_name]
[[<-StoragePath, path, p>] <String> Storage_path_name]
[[<-MountPoint, mount, mp, d>] <String> LUN_mount_point]
[[<-Force, f>] <SwitchParameter>
[[<CommonParameters>]]
```

Parameters

`-Server | -s` ***`Server_name`***

Specifies the host name or Microsoft failover cluster name of the host server from which to disconnect storage. The default is the local machine.

When the `Server_name` parameter is the name of a cluster, the cluster resource that is dependent on the connected LUN is removed first and then the LUN is disconnected from all the nodes of the cluster.

When a disk is connected as a CSV or as a shared disk to a cluster, the `Server` parameter must be a cluster name.

Required?

False

Position

Named

Default value

The local machine

Accept pipeline input?

False

Accept wildcard characters?

False

`-StoragePath` | **`-path`** ***`Storage_path_name`***

Specifies the full path of the LUN that you want to remove, including the LUN name, volume name, and storage system name. You must use either the storage path or the mount point.

Required?

False

Position

Named

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-Mountpoint` | **`-mount`** ***`LUN_mount_point`***

Specifies the mount point of the LUN that you want to remove.

Required?

False

Position

Named

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-Force` | **`-f`**

Specifies whether to forcefully disconnect the LUN when the file system in the LUN is in use, has virtual machines, or has open files. If you do not use this parameter, the file system cannot be unmounted due to the open file handles and, therefore, the LUN is not disconnected. The default is **`False`**. Use of the `Force` parameter is always required for clustered LUNs or when the `Server` parameter is a remote server name.

Required?

False

Position

Named

Default value

False

Accept pipeline input?

False

Accept wildcard characters?

False

Example: Removing storage

The following cmdlet unmounts the disk on mount point *M*, disconnects *server1* from the LUN on *storagesystemA* using the path */vol/vol1/lun1*, and deletes the LUN from *storagesystemA*:

```
Remove-OCStorage -Server server1 -StoragePath storagesystemA:/vol/
vol1/lun1 -Mountpoint M:\
```

Set-OCStorageSize cmdlet

The `Set-OCStorageSize` cmdlet enables you to expand or shrink the LUN and its underlying Windows volume and partition by a value that you specify. To expand the LUN, you specify a positive number. To shrink the LUN, you specify a negative number.

Syntax

```
Set-OCStorageSize
[[<-Server, s>] <String> server_name]
[[<-StoragePath, path, p> <String> storage_path_name]
[[<-MountPoint, mount, mp, d>] <String> LUN_Mount_point]
[[<-Size, sz, z>] <SwitchParameter> KB | MB | GB | TB
[[<CommonParameters>]]
```

Parameters

-Server | -s *Server_name*

Specifies the host name or Microsoft failover cluster name of the host server from which to connect storage.

When the *Server_name* parameter is the name of a cluster, the LUN is connected to all nodes of the cluster and the disk is configured as shared in a Windows failover cluster. The default is the local machine.

Required?

True

Position

0

Default value

The local machine

Accept pipeline input?

False

Accept wildcard characters?

False

`-StoragePath | -path | -p` ***`Storage_path_name`***

Specifies the full LUN path, including LUN name, volume name, and storage system name.

Required?

True

Position

1

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-MountPoint | -mount | -mp | -d` ***`LUN_mount_point`***

Specifies the LUN mount point. The default is the volume GUID. You can use a drive letter or a drive letter and directory as a mount point. A specified mount point must exist for a mounted LUN.

Required?

False

Position

2

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-Size |-z` `KB|MB|GB|TB`

Specifies the positive or negative size option for the resized LUN. The default unit is **`GB`**.

Required?

False

Position

3

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

Example: Resizing storage

The following cmdlets increase the Windows volume and the underlying LUN by the specified size of 1 GB:

```
Set-OCStorageSize -storagepath rishi:/vol/vol1/lun1 -verbose -z +1gb
Set-OCStorageSize -MountPoint M:\ -verbose -z +1gb
```

The following cmdlets decrease the windows volume and the underlying LUN by the specified size of 1 GB:

```
Set-OCStorageSize -Storagepath rishi:/vol/vol1/lun1 -verbose -z -1gb
Set-OCStorageSize -MountPoint M:\ -Verbose -z -1gb
```

Remove-OCVM cmdlet

The `Remove-OCVM` cmdlet deletes the specified virtual machines and associated VHDs on a local host or local cluster. Space is not reclaimed. The cmdlet also does not delete the resource group or any of your underlying storage or LUNs.

This cmdlet deletes the specified virtual machines and associated VHDs on a local host or local cluster.

Syntax

```
Remove-OCVM
[[<-VMName>] <String> Virtual_machine_name]
[[<-Port>] <String> Port_number
[[<CommonParameters>]]
```

Parameters

`VMName` | `-vmname` *`Virtual_machine_name`*

Specifies the name of the virtual machine that you want to delete.

Required?

True

Position

0

Default value

The local machine

Accept pipeline input?

False

Accept wildcard characters?

False

`-Port` | `-port` *`Port_number`*

Specifies the web service port number that is used for communication with VIMServiceHost.

Required?

False

Position

1

Default value

808

Accept pipeline input?

False

Accept wildcard characters?

False

Example

The following cmdlet deletes only the virtual machine, *`test`*, from the local host:

```
Remove-OCVM -VMName test -Verbose
```

Provisioning activities

You use the provisioning activities to create workflows to perform faster and more space-efficient provisioning activities such as provision, resize, and delete storage on your Hyper-V virtual machines with Microsoft System Center Orchestrator.

Connect Storage to Host activity

You can use the **Connect Storage to Host** activity to mount and connect LUNs that were previously on the host. The activity creates an igroup for the FCP initiator if one does not already exist and creates a session if one is not already active. You can use either the FC or iSCSI protocol to connect and map the LUN to multiple nodes of a cluster.

Properties

The following table indicates the information that you can enter in the properties tab for the **Connect Storage to Host** activity:

Property name	Type	Required?	Default
Storage path [1]	String	Required	None
Initiator name [2]	String	Optional	None
Is target storage Cluster Shared Volume [3]	Boolean	Optional	False
Resource group [4]	String	Optional	None
Resource name	String	Optional	None
Mount point [5]	String	Optional	None
Target server name [6]	String	Optional	localhost

1. Indicates the storage path, formatted as <*`StorageSystem:LunPath`*> .
2. Indicates one or more initiator names to which to map the LUN. If you do not specify an initiator and there is a Fibre Channel connection that is enabled between the host and the storage controller, the Fibre Channel initiator name is used. If a Fibre Channel initiator name is not used, either the first available iSCSI initiator with an established session is used or a new iSCSI session is created.
3. If the storage you want to connect to is a Cluster Shared Volume, you should set the property to **`True`**.

4. Indicates which resource group the storage belongs to after it is connected.
5. Indicates the mountpoint of the LUN. The mount point can be a drive letter, file path, or a volume GUID.
6. Indicates the server to which to connect the storage.

Published data

The following table indicates the information that the plug-in can return from the **Connect Storage to Host** activity:

Name	Type
CSV owner node name	String
Device ID	String
Disk cluster resource group name	String
Disk cluster resource name	String
Disk resource name	String
Disk resource owner node name	String
Disk serial number	String
Is CSV disk [1]	Boolean
Is cluster disk [2]	Boolean
Mount point	String
SCSI address	String
Storage path	String
Storage size [3]	String
Volume GUID	String

1. Returns `True` if the storage is a cluster shared volume; otherwise, it returns `False`.
2. Returns `True` if the storage is a cluster disk; otherwise, it returns `False`.
3. Returns the size, in bytes.

Disconnect Storage activity

You use the **Disconnect Storage** activity to disconnect a specific LUN from a host when it is no longer needed.

Properties

The following table indicates the information that you can enter in the properties tab for the **Disconnect Storage** activity:

Property name	**Type**	**Required?**	**Default**
Storage path [1]	String	Required	None
Mount point [2]	String	Required	None
Force disconnect [3]	Boolean	Optional	False
Target server name [4]	String	Optional	localhost

1. Indicates the storage path, formatted as <*`StorageSystem:LunPath`*> .
2. Indicates the mount point of the storage you want to disconnect. The mount point and the storage path cannot both be empty. You can use the volume GUID.
3. If there are FCP or iSCSI sessions, use **`True`** to force the system to disconnect.
4. Indicates the server from which you will disconnect the storage

Published data

The following table indicates the information that the plug-in can return from the **Connect Storage** activity:

Name	**Type**
CSV owner node name	String
Device ID	String
Disk cluster resource group name	String
Disk cluster resource name	String
Disk resource name	String
Disk resource owner node name	String
Disk serial number	String
Is CSV disk [1]	Boolean

Name	Type
Is cluster disk [2]	Boolean
Mount point	String
SCSI address	String
Storage path	String
Storage size [3]	String
Volume GUID	String

1. Returns `True` if the storage is a cluster shared volume; otherwise, it returns `False`
2. Returns `True` if the storage is a cluster disk; otherwise, it returns `False`
3. Returns the size, in bytes

Provision Storage activity

You use the **Provision Storage** activity to create a new LUN and connect it to a specified host.

Properties

The following table indicates the information that you can enter in the properties tab for the **Provision Storage** activity:

Property name	Type	Required?	Default
Storage path [1]	String	Required	None
Size [2]	String	Required	1
Size unit [3]	Choice	Required	GB
Initiator name [4]	String	Optional	None
Is target storage Cluster Shared Volume [5]	Boolean	Optional	False
Resource group [6]	String	Optional	None
Resource name	String	Optional	None
Mount point [7]	String	Optional	None
Target server name [8]	String	Optional	localhost

1. Indicates the storage path, formatted as <*`StorageSystem:LunPath`*>
2. Indicates the size of the LUN that you are creating.

3. Indicates the unit of measurement, such as KB, MB, GB, or TB.

4. Indicates one or more initiator names to which to map the LUN. If you do not specify an initiator and there is a Fibre Channel connection that is enabled between the host and the storage controller, the Fibre Channel initiator name is used. If a Fibre Channel initiator name is not used, either the first available iSCSI initiator with an established session is used or a new iSCSI session is created.

5. To provision the storage as a cluster shared volume, set to **`True`**.

6. Indicates which resource group the storage belongs to after it is connected.

7. Indicates the mountpoint of the LUN. The mount point can be a drive letter, file path, or a volume GUID.

8. Indicates the server with which the VIM web service server should communicate.

Published data

The following table indicates the information that the plug-in can return from the **Provision Storage** activity:

Name	**Type**
CSV owner node name	String
Device ID	String
Disk cluster resource group name	String
Disk cluster resource name	String
Disk resource name	String
Disk resource owner node name	String
Disk serial number	String
Is CSV disk [1]	Boolean
Is cluster disk [2]	Boolean
Mount point	String
SCSI address	String
Storage path	String
Storage size [3]	String
Volume GUID	String

1. Returns `True` if the storage is a cluster shared volume; otherwise, it returns `False`

2. Returns `True` if the storage is a cluster disk; otherwise, it returns `False`
3. Returns the size, in bytes.

Resize Storage activity

You can use the **Resize Storage** activity to increase or decrease the size of the storage on the host side to provide more efficient use of your storage. If the volume is full, the resize storage process fails.

Properties

The following table indicates the information that you can enter in the properties tab for the **Resize Storage** activity:

Property name	Type	Required?	Default
Storage path [1]	String	Required	None
Resize type [2]	Choice	Required	Expand
Size [3]	String	Required	1
Size unit [4]	Choice	Required	GB
Mount point [5]	String	Required	None
Target server name [6]	String	Optional	localhost

1. Indicates the storage path, formatted as <*`StorageSystem:LunPath`*>
2. Can be either "expand" or "shrink"
3. Indicates the size of the LUN that you are creating
4. Indicates the unit of measurement, such as KB, MB, GB, or TB
5. Indicates the mountpoint of the LUN. The mount point can be a drive letter, file path, or a volume GUID
6. Indicates the server to which the storage is connected

Published data

The following table indicates the information that the plug-in can return from the **Resize Storage** activity. The **Resize Storage** activity also adjusts the NTFS format to represent the changes to the storage.

Name	Type
CSV owner node name	String

Name	Type
Device ID	String
Disk cluster resource group name	String
Disk cluster resource name	String
Disk resource name	String
Disk resource owner node name	String
Disk serial number	String
Is CSV disk [1]	Boolean
Is cluster disk [2]	Boolean
Mount point	String
SCSI address	String
Storage path	String
Storage size [3]	String
Volume GUID	String

1. Returns `True` if the storage is a cluster shared volume; otherwise, it returns `False`
2. Returns `True` if the storage is a cluster disk; otherwise, it returns `False`
3. Returns the size, in bytes

Delete Storage activity

You use the **Delete Storage** activity to delete a specified LUN when it is no longer needed. To delete the LUN, you must have a storage path, a mount point, or a volume GUID.

Properties

The following table indicates the information that you can enter in the properties tab for the **Delete Storage** activity:

Property name	Type	Required?	Default
Storage path [1]	String	Required	None
Mount point [2]	String	Required	None
Force [3]	Boolean	Optional	None
Target server name [4]	String	Optional	localhost to the VIM web service server

1. Indicates the storage path, formatted as <*StorageSystem:LunPath*>
2. Indicates the mountpoint of the LUN. The mount point can be a drive letter, file path, or a volume GUID
3. Indicates that the storage be deleted even if the storage is in use. If **`True`**, the storage is deleted even if there is an active host connection or LUN map.
4. Indicates the server from which to delete the storage

Published data

The following table indicates the information that the plug-in can return from the **Delete Storage** activity:

Name	Type
CSV owner node name	String
Device ID	String
Disk cluster resource group name	String
Disk cluster resource name	String
Disk resource name	String
Disk resource owner node name	String
Disk serial number	String
Is CSV disk [1]	Boolean
Is cluster disk [2]	Boolean
Mount point	String
Initiator name [3]	String
Storage path	String
Storage size [4]	String
Volume GUID	String

1. Returns `True` if the storage is a cluster shared volume; otherwise, it returns `False`
2. Returns `True` if the storage is a cluster disk; otherwise, it returns `False`
3. Indicates one or more initiator names to which to map the LUN. If you do not specify an initiator and there is a Fibre Channel connection that is enabled between the host and the storage controller, the Fibre Channel initiator name is used. If a Fibre Channel initiator name is not used,

either the first available iSCSI initiator with an established session is used or a new iSCSI session is created.

4. Returns the size, in bytes

List Storage activity

You use the **List Storage** activity to discover LUNs and their mount points on the host so that you have a clear view of your storage.

Properties

The following table indicates the information that you can enter in the properties tab for the **List Storage** activity:

Property name	Type	Required?	Default
Storage path [1]	String	Optional	None
Mount point [2]	String	Optional	None
Target server name [3]	String	Optional	localhost

1. Indicates the storage path, formatted as <*`StorageSystem:LunPath`*>; can also be the mount point or volume GUID
2. Indicates the mount point of the LUN for which you want to create the list
3. Indicates the server to disconnect the storage from

Published data

The following table indicates the information that the plug-in can return from the **List Storage** activity:

Name	Type
CSV owner node name	String
Device ID	String
Disk cluster resource group name	String
Disk cluster resource name	String
Disk resource name	String
Disk resource owner node name	String
Disk serial number	String
Is CSV disk [1]	Boolean

Name	**Type**
Is cluster disk [2]	Boolean
Mount point	String
SCSI address	String
Storage path	String
Storage size [3]	String
Volume GUID	String

1. Returns `True` if the storage is a cluster disk; otherwise, it returns `False`
2. Returns `True` if the storage is a cluster disk; otherwise, it returns `False`
3. Returns the size, in bytes

Delete VM on Host activity

You use the **Delete VM on Host** activity to remove a previously cloned virtual machine. You can perform this operation only on the local Hyper-V host that has the VIM web service installed. Remote operations are not supported.

Properties

The following table indicates the information that you can enter in the properties tab for the **Delete VM on Host** activity:

Property name	**Type**	**Required?**	**Default**
VM name [1]	String	Required	None

1. Indicates the name of the virtual machine to delete.

Published data

The following table indicates the information that the plug-in can return from the **Delete VM on Host** activity:

Name	**Type**	**Format**
Deleted VM name	String	

Cloning cmdlets and activities

You can use the PowerShell cloning cmdlets and Orchestrator activities with Microsoft System Center Virtual Machine Manager (SCVMM) applications to manage your clones and to perform a variety of cloning tasks to improve performance and space efficiency on your Hyper-V virtual machines.

Cloning storage with OCPM

By using cloning cmdlets with Microsoft System Center Virtual Machine Manager (SCVMM) applications, you can use your existing storage resources to perform faster and more efficient use of the space on your Hyper-V virtual machines.

New-OCClone cmdlet

The `New-OCClone` cmdlet clones Hyper-V virtual machines that are currently in an off state. You can clone such a virtual machine either from an existing virtual machine, by specifying the source virtual machine name or GUID, or from a System Center Virtual Machine Manager (SCVMM) template. `New-OCClone` supports clustered Data ONTAP.

You can then deploy the cloned virtual machines to the specified host system or server.

Syntax

```
New-OCClone
[[<-Server, s>] <String> Server_name]
[[<-VMMServer, vmms>] <String> VMM_server_name]
[[<-Template, tm>] <String> Template_name]
[[<-VMName, vm>] <String> VM_Name]
[[<-VMGUID, guid} <String> VM_GUID]
[[<-BaseVMName, bv>] <String> Virtual_machine_name]
[[<-NumberofClones, nc>] <Int32> Number_of_clones]
[[<-MountPoint, mount, mp, d>] <String> Mount_point_location]
[[<-AllInSingleLUN, asnl>] <Boolean> True | False]
[[<-AllinSourceLUN, asrl>] <Boolean> True | False]
[[<-JustCloneVHD, jcl>] <Boolean> True | False]
[[<-StartVM, svm>] <Boolean> True | False]
[[<-ClusterSharedVolume, csv <Boolean> True | False]
[-XenImport, xen>] <SwitchParameter>]
[[<CommonParameters>]]
```

Description

Cloning an existing virtual machine and cloning a template require different input parameters and support slightly different operations.

Cloning an existing virtual machine has the following requirements:

- Either a virtual machine name or the GUID of an existing virtual machine is required.
- The virtual machine must exist on the host where the cmdlet is run.
- The cloned virtual machines can be deployed to the same host or to a parent cluster of the host on which the cmdlet is run.
- You must have run sysprep (Microsoft's system preparation tool) on the VHDs of the existing virtual machine, and the virtual machine must not have been started after that.

Cloning from a template has the following requirements:

- An SCVMM server name and template name are required.
- The cloned virtual machines can be deployed to the same host on which the cmdlet is run or to its parent cluster.
- For clustered Data ONTAP only: only template cloning is supported (SMB to SMB and LUN to LUN).
- For clustered Data ONTAP only: SMB to LUN or LUN to SMB cloning is not supported.

Parameters

`-Server` | **`-s`** ***`Server_name`***

Specifies the name of the host or Microsoft failover cluster of the Hyper-V server on which the cloned virtual machine is to run.

When this parameter is the name of a cluster, the cloned virtual machines are highly available. If a new LUN is created for the clone, that LUN is connected to all nodes of the cluster, and the disk is configured as shared in a Windows failover cluster.

If the command clones into a source LUN or an existing LUN, and the **`Server_name`** parameter is the name of a cluster, the existing LUN must be a shared disk that is connected to all the nodes of the cluster.

The default is the host from which the command is run, or the cluster name if the host is a member of a cluster.

Required?

False

Position

0

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-VMMServer` | **`-vmms`** ***`VMM_server_name`***

Specifies the name of the SCVMM server on which the virtual machine template resides. This parameter is required when the `Template` parameter is specified.

Required?

False

Position

1

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-Template` | **`-tn`** ***`Template_name`***

Specifies the name of the SCVMM virtual machine template that you want to clone. This parameter is required when the `VMMServer` parameter is specified.

Required?

False

Position

2

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-VMName` | **`-vmname`** ***`VM_Name`***

Specifies the name of the virtual machine that you want to clone. If the name does not uniquely identify a particular virtual machine, the command returns an error. The source virtual machines and VHDs that you want to clone must already exist.

Either the `VMName` or `VMGUID` parameter is required to clone an existing virtual machine if either the `Template` or `VMMServer` parameter is not specified.

Required?

False

Position

3

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-VMGUID` | **`-vmguid`** ***`VM_GUID`***

Specifies the GUID of the virtual machine that you want to clone. When both the `VMGUID` and `VMName` parameter are provided, the`VMGUID` parameter is used to uniquely identify the virtual machine.

Either the `VMName` or `VMGUID` parameter is required to clone an existing virtual machine if neither the `Template` nor `VMMServer` parameters are specified.

Required?

False

Position

4

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-BaseVMName` | **`-bv`** ***`Virtual_machine_name`***

Specifies the base name of the cloned virtual machine.

Required?

False

Position

5

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-NumberofClones` | **`-nc`** ***`Number_of_clones`***

Specifies the number of clones that you want to create.

Required?

False

Position

6

Default value

1

Accept pipeline input?

False

Accept wildcard characters?

False

`-MountPoint` | **`-mount | -mp | -d`** ***`Mount_point_location`***

Specifies the mountpoint for an existing LUN or of a new LUN to which the new clones are mounted. This parameter is valid when the `AllInSingleLun` parameter is specified. If the `AllInSingleLun` parameter is not specified, but the`MountPoint` parameter is specified, the `AllInSingleLun` parameter is assumed.

The mountpoint could be in one of the following forms:

- *`<Driveletter:\>`*
- *`<Driveletter:\VolumeMountPointFolder>`*
- `VolumeGUID` for an existing mounted NTFS volume
- CSV Reparsepoint `c:\<`*`ClusterStorage`*`>\<`*`VolumeN`*`>`(only for an existing LUN as a clone destination)

Required?

False

Position

7

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-AllinSingleLUN` | **`-asnl`** **`True`** | **`False`**

Specifies that the cloning process is done in a new or existing single LUN and applies for both templates and virtual machines. If you specify the `MountPoint` parameter, the cloning process checks for the existence of the mountpoint and it is used as the destination for cloning.

If the mountpoint that you specified is not present on the target host, this command creates a new LUN, mounts it at the specified mountpoint, and creates the clones in that LUN.

If the mountpoint is not specified with this parameter, a new LUN is created and mounted using the volume GUID that is generated by the operating system.

When the `AllInSingleLUN` parameter is not specified, the presence of the `MountPoint` parameter provides the same behavior. When neither the `AllInSingleLUN`, the `MountPoint`, nor the `AllInSourceLun` parameters are specified, a new LUN is created and mounted using the volume GUID for each clone.

Required?

False

Position

Named

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-AllinSourceLUN` | **`-asrl`** **`True`** | **`False`**

Specifies that the clone or clones be created in the same source NTFS or CSV on the LUN. This parameter applies only for virtual machine cloning. The cmdlet returns an error if you specify the `AllInSourceLun` parameter along with the `Template` parameter.

When the `AllInSourceLUN` parameter is specified, neither the `AllInSingleLUN` nor the `MountPoint` parameters are needed and are ignored.

Required?

False

Position

Named

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-JustCloneVHD` | `-jcl` `True` | `False`

Specifies that the VHDs from the template or existing virtual machine are cloned and mounted on the specified server but that the virtual machine configuration is not created.

Required?

False

Position

Named

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-StartVM` | `-svm` `True` | `False`

Specifies whether you want to start the virtual machine or machines after they are created. The default is **`False`**. This parameter is not valid and is ignored when you use it with the `JustCloneVHD` parameter.

Required?

False

Position

Named

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

-ClusterSharedVolume |-csv True|False

Specifies whether the new LUNs that are created for the cloned virtual machines are added to the Cluster Shared Volume.

Required?

False

Position

Named

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

-XenImport|-xen

Specifies the path to the XenImport file, which is currently as follows:

```
C:\Users\rsu\AppData\Local\Temp
\xendesktop_4_0_2011_03_30_17_08.csv
```

Required?

False

Position

Named

Default value

False

Accept pipeline input?

False

Accept wildcard characters?

False

Examples: Creating new clones

The following cmdlet creates a new clone on the server `vmhost12`. The user-specified `AllInSourceLUN` parameter is allowed because the user did not specify a template.

```
New-OCClone -server vmhost12 -Vmname GoldVM1 -AllInSourceLUN -
BaseVMName finance-w2k3-oracle
```

The following cmdlet creates a new clone from a template `WinTemplate`. The parameter `AllInSingleLUN` is not necessary if the mountpoint already exists. The clone is placed into that mountpoint.

```
New-OCClone -Server vmhost12 -VmmServer VMMserver1 -Template
WinTemplate -BaseVMName finance-w2k3-oracle
```

New-OCCloneFile cmdlet

The `New-OCCloneFile` cmdlet clones any file on a mounted NTFS or cluster shared volume (CSV) on a Data ONTAP LUN to a specified path within another mounted NTFS or CSV on a Data ONTAP LUN. This cmdlet does not create the directory.

Syntax

```
New-OCCloneFile
[[<-SourceFile, sf>] <String> Source_File]
[[<-ClonedFile, cfl>] <String> Clone_File]
[<CommonParameters>]
```

Description

This cmdlet supports traditional windows LUNs or CSV LUNs within a Windows cluster.

To use the `New-OCCloneFile` cmdlet, your environment must meet the following requirements:

- The directory of the destination path must exist.
- Both the source and destination NTFS or CSV must be mounted on the same host.
- If the source or destination path is on a CSV, the CSV must be owned by the node from which the command is run.
- The existing NTFS or CSV must have enough space to hold the new cloned file.

Parameters

`-SourceFile` | `-sf` *`Source_File`*

Specifies the file to be cloned.

Required?

True

Position

0

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-CloneFile` | **`-cfl`** ***`Clone_File`***

Specifies the destination file path.

Required?

True

Position

1

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

Example: Cloning a file

The following cmdlet creates a clone of the file `w2k8-r2-sp1.vhd` that resides on `goldenvhds`. The clone is named `ab-test-vm-1.vhd` and stored on `ab-test-vm-1`.

```
New-OCCloneFile -SourceFile g:\goldenvhds\w2k8-r2-sp1.vhd -CloneFile
h:\ab-test-vm-1\ab-testvm-1.vhd
```

Cloning activities

By using cloning activities with Microsoft System Center Virtual Machine Manager (SCVMM) applications, you can use your existing storage resources to more efficiently use the space on your Hyper-V virtual machines on System Center Orchestrator.

Clone VM activity

You use the **Clone VM** activity to create multiple clones of virtual machines without an SCVMM template in the host. You cannot clone virtual machines on remote hosts. The source virtual machine must be in the offline state to complete this operation.

Requirements

You must meet the following prerequisites before using the **Clone VM** activity:

- The specified target server must be in either the same domain or in a trusted domain of the host on which you run the command.
  You cannot clone a virtual machine on a remote host.
- You must have an active iSCSI session or FCP connectivity from the specified target server to the storage system.
- You must have already added the credentials for the storage system.

Properties

The following table indicates the information that you can enter in the properties tab for the **Clone VM** activity:

Property name	**Type**	**Required?**	**Default**
Target server name	String	Required	localhost
VM name[1]	String	Required	None
Clone VM name prefix	String	Required	None
Clone only VHDs	Boolean	Optional	False
Is clone storage Cluster Shared Volumes [2]	Boolean	Optional	False
Is XenDesktop import [3]	String	Optional	None
Mount point to the cloned storage [4]	String	Optional	None
Number of VMs to clone	Integer	Optional	1
Place cloned VMs in a single LUN [5]	Boolean	Optional	False
Place cloned VMs in the source LUN [6]	Boolean	Optional	False
Start VM after clone	Boolean	Optional	False

Property name	Type	Required?	Default
VM GUID	String	Optional	None

1. The case of the letters in the existing virtual machine name in the Hyper-V manager should match that of the names in the host

2. If you want to connect the storage as CSVs, set the property to **`True`**

3. This property specifies the path to the XenImport file, which is currently `C:\Users\rsu \AppData\Local\Temp \xendesktop_4_0_2011_03_30_17_08.csv`

4. This property indicates the mount point to use for the cloned storage. If more than one clone is specified, only the first clone is connected to the mount point

5. When this property is `True`, the virtual machine is cloned in a new LUN. If the number of clones is greater than **`1`**, all the virtual machines are cloned in that new LUN. When `False`, each virtual machine is cloned in a new LUN. You can set this to `True` only when the `Place cloned VMs in source LUN` property is `False`.

6. When this property is `True`, the virtual machine is created in the source LUN. When this property is `False`, each virtual machine is cloned in a new LUN. You can set this to `True` only when the `Place cloned VMs on a single LUN` property is `False`.

Published data

The **Clone VM** object publishes six data entries: either a list of virtual machine clone objects or a list of VHD clone objects. However, only the contents of four data entries are visible, based on the setting of the `Clone only VHDs` property:

- If the `Clone only VHDs` property is `True`, then the `Cloned VM Name` property and the `VM path` property do not contain any data.

- If the `Clone only VHDs` property is `False`, the `VHD Path` property and the `VHD size in byte` property do not contain any data.

The following table indicates the information that the plug-in can return from the **Clone VM** object:

Name	Type
Cloned VM name [1]	List of virtual machine clone objects
Storage path [1]	
Target host name [1]	
VM path [1]	

Name	**Type**
Storage path [2]	List of VHD clone objects
Target host name [2]	
VHD path [2]	
VHD size in bytes [2]	

1. Indicates the output when the Clone Only VHDs property is `False`
2. Indicates the output when the Clone Only VHDs property is `True`

Clone VM from SCVMM Template activity

You use the **Clone VM from SCVMM Template** activity to clone virtual machines from a System Center Virtual Machine Manager (SCVMM) template. You can use this activity to create multiple clones. After the clones are created, the virtual machines are mapped to the destination host.

Properties

The following table indicates the information you can enter in the properties tab for the **Clone VM from SCVMM Template** activity:

Property name	**Type**	**Required?**	**Default**
Target server name	String	Required	localhost
SCVMM server name [1]	String	Required	localhost
SCVMM template name	String	Required	None
Clone VM name prefix	String	Required	None
Clone only VHDs	Boolean	Optional	False
Is the clone storage a cluster shared volume	Boolean	Optional	False
Is XenDesktop import [2]	String	Optional	None
Mount point to the cloned storage [3]	String	Optional	None
Number of VMs to clone	Integer	Optional	1
Place cloned VMs in a single LUN	Boolean	Optional	False
Start VM after clone	Boolean	Optional	False

1. The name of the host on which the SCVMM server is installed.
2. This property specifies the path to the XenImport file, which is currently `C:\Users\rsu\AppData\Local\Temp \xendesktop_4_0_2011_03_30_17_08.csv`.
3. This property indicates the mount point to use for the clone storage. If more than one clone is specified, only the first clone is connected to the mount point.

Published data

The **Clone VM from SCVMM Template** activity publishes six data entries: either a list of virtual machine clone activities or a list of VHD clone activities. However, only the contents of four entries are visible, based on the setting of the `Clone only VHDs` property:

- If the `Clone only VHDs` property is **`True`**, then the `Cloned VM Name` property and the `VM path` property do not contain any data.
- If the `Clone only VHDs` property is **`False`**, then the `VHD Path` property and the `VHD size in byte` property do not contain any data.

The following table indicates the information that the plug-in can return from the **Clone VM from SCVMM Template** activity:

Name	**Type**	**Format**
Cloned VM name [1]	List of virtual machine clone objects	
Storage path [1]		
Target host name [1]		
VM path [1]		
Storage path [2]	List of VHD clone objects	
Target host name [2]		
VHD path [2]		
VHD size in bytes [2]		

1. Output when the Clone Only VHDs property is `False`.
2. Output when the Clone Only VHDs property is `True`.

Clone NTFS File activity

You use the **Clone NTFS File** activity to create an exact replica of a single file. Both the source file and the cloned VHD reside on the same volume.

Properties

The following table indicates the information that you can enter in the properties tab for the **Clone NTFS File** activity:

Property name	Type	Required?	Default
Target server name	String	Required	localhost
Source file path [1]	String	Required	None
Clone file path [2]	String	Required	None

1. Indicates the full path to the source file that you want to clone.
2. Indicates the full path to the destination of the clone.

Published data

The following table indicates the information that the plug-in can return from the **Clone NTFS File** activity:

Name	Type	Format
Cloned file path [1]	String	

1. Indicates the path to the cloned file.

Managing crash-consistent disaster recovery using cmdlets and activities

Crash-consistent disaster recovery uses the SnapMirror feature to replicate data across primary and secondary sites following a disaster.

To properly implement disaster recovery, you must be familiar with the SnapMirror feature, the requirements for disaster recovery, and several disaster recovery cmdlets and activities, including composite cmdlets and automated activities and their properties.

Disaster recovery cmdlets

You can use disaster recovery cmdlets to control failover and failback procedures, make changes to the disaster recovery plan, and control the SnapMirror relationships. Disaster recovery cmdlets perform a single operation but can be used together in a *composite activity*, to complete a more complex process.

Related concepts

Managing crash-consistent disaster recovery using cmdlets and activities on page 101

New-OCDRPlan cmdlet

The `New-OCDRPlan` cmdlet creates a new disaster recovery plan, which describes disaster recovery objects and determines how they behave during failover and failback.

Syntax

```
New-OCDRPlan
[[<-PrimaryServer, prisvr>] <String> Primary_server_name]
[[<-SecondaryServer, secsvr>] <String> Secondary_server_name]
[[<-DRPlanName, drn>] <String> DRPlan_name]
[[<-DRPlanFolder, drf>] <String> Name_of_folder_for_DRPlan_file]
[[<-Failback, fb>] <Boolean>]
[[<-Port>] <String> Port_number]
[[<-CommonParameters>]]
```

Description

You can specify a file path and a file name for the plan. If the file path is not specified, then the plan is created in the OCPM program data folder: for example, `C:\ProgramData\OnCommand\MS_Plugin`.

If the file name of the plan is not specified, then a default plan name is created using the following convention:

`PrimaryServerOrCluster_SecondaryServerOrCluster_DRPlan.xml`

On Windows 2012 server platforms, there is no constraint to Turn Off the Hyper-V virtual machines. Disaster recovery plans are generated with live or running virtual machines. On Windows 2012, the virtual machines are not exported and their virtual machine configuration files are not generated. However, on Windows 2008 R2 server platforms, the virtual machines are exported and their virtual machine configuration files are generated along with the disaster recovery plan file.

Parameters

`-PrimaryServer |-prisvr` ***`Primary_server_name`***

Specifies the host name or cluster node name of the primary server.

Required?

True

Position

0

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-SecondaryServer |-secsvr` ***`Secondary_server_name`***

Specifies the destination host name or cluster node name of the secondary server.

Required?

True

Position

1

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-DRPlanName |-drn` ***`DRPlan_name`***

Specifies the file name of the plan.

Required?

False

Position

2

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-DRPlanFolder | -drf` ***`Name_of_folder_for_DRPlan_file`***

Specifies the folder name of the plan. The default is the program data location for disaster recovery configuration files: `C:\ProgramData\<product_folder>`.

For common access to a network resource, such as a file that is shared across the primary and secondary sites, you should use network UNC paths with proper share privileges for the *`DRPlanFolder/DRPlan`* paths or virtual machine configuration files. UNC paths are expressed as `\\ComputerName\SharedFolder`.

Required?

False

Position

3

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-Failback | -fb`

Specifies the disaster recovery operation type: failover or failback.

Required?

False

Position

4

Default value

Failover

Accept pipeline input?

False

Accept wildcard characters?

False

`-Port` | **`-port`** ***`Port_number`***

Specifies the web service port number that is used for communication with VIMServiceHost windows service.

Required?

False

Position

5

Default value

808

Accept pipeline input?

False

Accept wildcard characters?

False

Example: Creating a new disaster recovery plan

The following cmdlet creates a disaster recovery plan for a clustered primary server, *`SFO_Cluster`*, and a clustered secondary server, *`NY_Cluster`*:

`New-OCDRPlan -PrimaryServer` ***`SFO_Cluster`*** **`-SecondaryServer`** ***`NY_cluster`*** **`-Verbose`**

The `-DRPlanFolder` and `-DRPlanName` parameters are not specified, so the plan is named `SFO_NY_DRPlan.xml` and is created in the default folder, `C:\ProgramData\OnCommand\MS_Plugin\SFO_NY_DRPlan.xml`, on the primary server.

Confirm-OCDRPlan cmdlet

The `Confirm-OCDRPlan` cmdlet validates the current state of either the primary storage system or the secondary storage system, based on the information in the specified disaster recovery plan. If the validation fails, the plan is not current, and you must run the `Update-OCDRPlan` cmdlet.

Syntax

```
Confirm-OCDRPlan
[[<-DRPlan, drp>] <String> DR_path_with_DRPlan_file_name]
```

```
[[<-Port>] <String> Port_number]
[[<-CommonParameters>]]
```

Parameters

`-DRPlan` | `-drp` *`DRPlan_path_with_file_name`*

Specifies the disaster recovery path, including the disaster recovery plan file name.

Required?

True

Position

0

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-Port` | `-port` *`Port_number`*

Specifies the web service port number that is used for communication with VIMServiceHost windows service.

Required?

False

Position

1

Default value

808

Accept pipeline input?

False

Accept wildcard characters?

False

Example: Validating the SnapMirror relationships against the disaster recovery plan

The following cmdlet validates SnapMirror relationships to the secondary site and verifies that there are no conflicts on the secondary site based on the existing disaster recovery plan that is located at `\\`*`Server1\DRPlans\SFO_NY_DRPlan.xml`*:

```
Confirm-OCDRPlan -DRPlan \\Server1\DRPlans\SFO_NY_DRPlan.xml -Verbose
```

Update-OCDRPlan cmdlet

The `Update-OCDRPlan` cmdlet updates and validates the disaster recovery plan on either the primary site, the secondary site, or both, with the latest configuration information. Whenever the primary or secondary site configuration changes, you must update and validate the disaster recovery plan. If the validation fails, the plan is not up-to-date and must be refreshed.

Syntax

```
Update-OCDRPlan
[[<-DRPlan, drp>] <String> DRPlan_path_with_file_name]
[[<-PrimaryServer, prisvr>] <String> Primary_server_name]
[[<-SecondaryServer, secsvr>] <String> Secondary_server_name]
[[<-Failback, fb>] <SwitchParameter>]
[[<-Port>] <String> Port_number]
[[<-CommonParameters>]]
```

Parameters

-DRPlan | -drp *DRPlan_path_with_file_name*

Specifies the disaster recovery path, including the disaster recovery plan file name.

Required?

False

Position

0

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

-PrimaryServer | -prisvr *Primary_server_name*

Specifies the host name or cluster node name of the primary server.

Required?

False

Position

1

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-SecondaryServer | -secsvr` ***`Secondary_server_name`***

Specifies the destination host name or cluster node name of the secondary server.

Required?

False

Position

2

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-Failback | -fb`

Specifies the disaster recovery operation type: either failover or failback.

Required?

False

Position

3

Default value

Failover

Accept pipeline input?

False

Accept wildcard characters?

False

`-Port | -port` ***`Port_number`***

Specifies the web service port number that is used for communication with VIMServiceHost windows service.

Required?

False

Position

4

Default value

808

Accept pipeline input?

False

Accept wildcard characters?

False

Example: Updating the disaster recovery plan

The following cmdlet updates an existing disaster recovery plan, *`SFO_NY_DRPlan.xml`*, with the latest information from the specified primary and secondary server:

```
Update-OCDRPlan -DRPlan C:\ProgramData\OnCommand\MS_Plugin
\SFO_NY_DRPlan.xml -PrimaryServer SFO_Cluster -SecondaryServer
NY_cluster -Verbose
```

Connect-OCDRLUN cmdlet

The `Connect-OCDRLUN` cmdlet connects all the LUNs in the disaster recovery plan to the secondary host or cluster, as specified in the disaster recovery plan.

Syntax

```
Connect-OCDRLUN
[[<-DRPlan>] <String> DRPlan_path_with_file_name]
[[<-Force>] <SwitchParameter>]
[[<-Port>] <String>-Port_name]
[[<-CommonParameters>]]
```

Parameters

`-DRPlan` | `-drp` *`DRPlan_path_with_file_name`*

Specifies the disaster recovery path to the plan file that is specified by name.

Required?

True

Position

0

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-Force | -f`

Specifies whether to forcefully disconnect conflicting mount points and break the SnapMirror relationships.

Required?

False

Position

1

Default value

False

Accept pipeline input?

False

Accept wildcard characters?

False

`-Port | -port`

Specifies the web service port number that is used for communication with VIMServiceHost windows service.

Required?

False

Position

2

Default value

808

Accept pipeline input?

False

Accept wildcard characters?

False

Example: Connecting LUNs on the disaster recovery site

The following cmdlet connects all the LUNs to the disaster recovery site based on the disaster recovery plan that is located at *`\\Server1\DRPlans\SFO_NY_DRPlan.xml`*. After this operation finishes successfully, all the necessary LUNs are connected on the disaster recovery site.

Connect-OCDRLUN -DRPlan \\*Server1*\DRPlans\SFO_NY_DRPlan.xml -Verbose -Force

Initialize-OCDRMirror cmdlet

This cmdlet begins the initial complete (baseline) SnapMirror Snapshot copy replication from a source volume or qtree to a destination volume or qtree. You must use this cmdlet to perform a complete transfer of information whenever you start up a SnapMirror source-destination relationship for the first time.

Syntax

```
Initialize-OCDRMirror
[[<-DestinationLocation, destloc>] <String> Destination_location]
[[<-DestinationSnapshot, destsnap>] <String>] Destination_SnapShot
[[<-MaxTransferRate, maxrate>] <String> KBPS]
[[<-SourceLocation, srcloc>] <String> Source_location]
[[<-SourceSnapshot, srcsnap>] <String> Source_SnapShot]
[[<-Port>] <String> Port_number]
[[<-CommonParameters>]]
```

Description

Mirror relationship initializations are handled asynchronously, so you must make sure that a transfer is complete before you execute additional operations.

You can use the `Get-OCDRMirrorsStatus` cmdlet or web service to get the status of operations. You can periodically poll the system with this service to determine when the transfer is complete.

This cmdlet is run on the Hyper-V host, and that host must contain the credentials for both the primary and secondary controllers. This cmdlet also requires a destination location for the replication operation.

Parameters

-DestinationLocation | -destloc *Destination_location*

Specifies the destination location of the SnapMirror pair. The destination location can be expressed in either of the following forms:

- The volume form:

  <filer>:<volume>

- The qtree form:

`<filer>:/vol/<volume>/<qtree>`

The **`<filer>`** value must match the destination storage system. If the `DestinationLocation` parameter is in the volume form, the volume must be in the restricted state. If the `-DestinationLocation` parameter is in the qtree form, the qtree must not already exist.

Required?

True

Position

0

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-DestinationSnapshot` | `-destsnap`

Specifies that the destination Snapshot copy (in addition to the regular SnapMirror Snapshot copy) on the secondary site should be created after the qtree SnapMirror transfer is complete.

Required?

False

Position

1

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-MaxTransferRate` | `-maxrate`

Specifies the maximum transfer rate, in kilobytes per second (KBps). The default is the highest transfer speed at which the storage system can transfer data.

Required?

False

Position

2

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-SourceLocation` | `-srcloc` *`Source_location`*

Specifies the source location of the initial transfer. The source location can be expressed in either of the following forms:

- The volume form:

  `<filer>:<volume>`

- The qtree form:

  `<filer>:/vol/<volume>/<qtree>`

  If the `SourceLocation` parameter is not specified, the source in `/etc/snapmirror.conf` is used for the destination path.

Required?

False

Position

3

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-SourceSnapshot` | `-srcsnap`

Specifies which source Snapshot copy to use for a qtree update. If the `Source-Snapshot` parameter is not specified, a new Snapshot copy is created.

Required?

False

Position

4

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-Port` | `-Port_number`

Specifies the web service port number that is used for communication with VIMServiceHost windows service.

Required?

False

Position

5

Default value

808

Accept pipeline input?

False

Accept wildcard characters?

False

Example: Initializing SnapMirror relationships

The following cmdlet initializes the SnapMirror relationships across the storage controllers *`Primary-Site`* and *`Secondary-Site`*, and between the volumes *`SourceVolume`* and *`DestinationVolume`*. The sizes of the destination volumes must be greater than or equal to the sizes of the source volumes for SnapMirror to initialize. The destination Data ONTAP version of the storage controller must be the same as the Data ONTAP version of the source storage controller.

```
Initialize-OCDRMirror -SourceLocation Primary-Site:SourceVolume -
DestinationLocation Secondary-Site:DestinationVolume -Verbose
```

Invoke-OCDRMirrorStop cmdlet

The `Invoke-OCDRMirrorStop` cmdlet cancels any data transfer to a specified destination based on information in the disaster recovery plan. This cmdlet can be run on the Hyper-V host, which must contain local storage system credentials for both the primary and secondary controllers.

Syntax

```
Invoke-OCDRMirrorStop
[[<-DRPlan, drp>] <String> DRPlan_path_with_file_name]
[[<-ClearCheckpoint, ccp>] <String> True | False]
[[<-DestinationLocation>] <String> Destination_location_name]
[[<-Port>] <String> Port_number]
[[<-CommonParameters>]]
```

Parameters

`-DRPlan` | `-drp` ***`DRPlan_path_with_file_name`***

Specifies the disaster recovery path by using the disaster recovery plan file name.

Required?

True

Position

0

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-ClearCheckpoint` | `-ccp True | False`

If **`True`**, the restart checkpoint is cleared.

Required?

False

Position

1

Default value

False

Accept pipeline input?

False

Accept wildcard characters?

False

`-DestinationLocation` | `-destloc` *`Destination_location_name`*

Specifies the destination of the transfer. The destination should be expressed in either of the following forms:

- The volume form:

  `<filer>:<volume>`

- The qtree form:

  `<filer>:/vol/<volume>/<qtree>`

Required?

False

Position

2

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-Port` | `-port`

Specifies the web service port number that is used for communication with VIMServiceHost windows service.

Required?

False

Position

3

Default value

808

Accept pipeline input?

False

Accept wildcard characters?

False

Example: Canceling a data transfer

The following cmdlet cancels the data transfer based on information in the disaster recovery plan *Sunnyvale_Seattle_DRPlan.xml:*:

```
Invoke-OCDRMirrorStop -DRPlan "C:\ProgramData\OnCommand\MS_Plugin
\Sunnyvale_Seattle_DRPlan.xml" Port_number -Verbose
```

Invoke-OCDRMirrorBreak cmdlet

The `Invoke-OCDRMirrorBreak` cmdlet breaks all the SnapMirror relationships on the secondary storage system based on information in the disaster recovery plan. If any of the SnapMirror relationships are in an unknown state, the cmdlet returns an error.

Syntax

```
Invoke-OCDRMirrorBreak
[[<-DRPlan, drp>] <String> DRPlan_path_with_file_name]
[[<-ClearCheckpoint, ccp>] <String> True | False]
[[<-DestinationLocation, destloc>] <String> Destination_location]
[[<-Port>] <String> Port_number]
[[<-CommonParameters>]]
```

Description

The cmdlet must be issued on the Hyper-V host, and that host must contain the credentials for both the primary and secondary controllers.

Parameters

-DRPlan | **-drp** ***DRPlan_path_with_file_name***

Specifies the disaster recovery path by using the disaster recovery plan name.

Required?

True

Position

0

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-ClearCheckpoint` | **`-ccp`** **`True | False`**

If **`True`**, the restart checkpoint is cleared.

Required?

False

Position

1

Default value

False or not cleared

Accept pipeline input?

False

Accept wildcard characters?

False

`-DestinationLocation` | **`-destloc`** ***`Destination_location`***

Specifies the destination of the mirror copies for which you want to break the relationship.

The destination location should be expressed in either of the following forms:

- The volume form:

  `<filer>:<volume>`

- The qtree form:

  `<filer>:/vol/<volume>/<qtree>`

Required?

False

Position

2

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-Port` | **`-port`** ***`Port_number`***

Specifies the web service port number that is used for communication with VIMServiceHost windows service.

Required?

False

Position

3

Default value

808

Accept pipeline input?

False

Accept wildcard characters?

False

Example: Breaking SnapMirror relationships

The following cmdlet breaks the SnapMirror relationships on the secondary storage system based on information in the disaster recovery plan `Sunnyvale_Seattle_DRPlan.xml`:

```
Invoke-OCDRMirrorBreak -DRPlan C:\ProgramData\OnCommand\MS_Plugin
\Sunnyvale_Seattle_DRPlan.xml -Verbose
```

Invoke-OCDRMirrorQuiesce cmdlet

The `Invoke-OCDRMirrorQuiesce` cmdlet temporarily pauses (quiesces) mirror relationships based on information in the specified disaster recovery plan.

Syntax

```
Invoke-OCDRMirrorQuiesce
[[<-DRPlan, drp>] <String> DRPlan_path_with_file_name]
[[<-DestinationLocation, destloc>] <String> Destination_location]
[[<-Port>] <String> Port_number]
[[<-CommonParameters>]]
```

Description

All SnapMirror scheduled transfers are halted until you issue the `Invoke-OCDRMirrorResume` cmdlet, which must be issued on the Hyper-V host, and that host must contain the credentials for both the primary and secondary controllers.

Parameters

`-DRPlan | -drp DRPlan_path_with_file_name`

Specifies the disaster recovery path by using the disaster recovery plan name.

Required?

True

Position

0

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-DestinationLocation` | **`-destloc`** ***`Destination_location`***

Specifies the destination of the mirror relationships to quiesce. The destination should be expressed in either of the following forms:

- The volume form:

  `<filer>:<volume>`

- The qtree form:

  `<filer>:/vol/<volume>/<qtree>`

Required?

False

Position

1

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-Port` | **`-port`** ***`Port_number`***

Specifies the web service port number that is used for communication with VIMServiceHost windows service.

Required?

False

Position

2

Default value

808

Accept pipeline input?

False

Accept wildcard characters?

False

Example: Quiescing the SnapMirror relationships

The following cmdlet quiesces the SnapMirror relationships for the mirror copies that are captured in the disaster recovery plan *Sunnyvale_Seattle_DRPlan.xml*:

```
Invoke-OCDRMirrorQuiesce -DRPlan C:\ProgramData\OnCommand\MS_Plugin
\Sunnyvale_Seattle_DRPlan.xml -Verbose
```

Invoke-OCDRMirrorResume cmdlet

The `Invoke-OCDRMirrorResume` cmdlet resumes the mirror relationships based on information in the specified disaster recovery plan. Transfers that were previously paused (quiesced) resume to the specified destination. This cmdlet is run on the Hyper-V host, and that host must contain the credentials for both the primary and secondary controllers in its local storage system.

Syntax

```
Invoke-OCDRMirrorResume
[[<-DRPlan, drp>] <String> DRPlan_path_with_file_name]
[[<-DestinationLocation, destloc>] <String> Destination_location]
[[<-Port>] <String> Port_number]
[[<-CommonParameters>]]
```

Parameters

-DRPlan | -drp *DRPlan_path_with_file_name*

Specifies the disaster recovery path, including the disaster recovery plan file name.

Required?

True

Position

0

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-DestinationLocation` | `-destloc` *`Destination_location`*

Specifies the destination location of the mirrored elements for which you want to resume a relationship. The destination location should be expressed in either of the following forms:

- The volume form:

  `<filer>:<volume>`

- The qtree form:

  `<filer>:/vol/<volume>/<qtree>`

Required?

False

Position

1

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-Port` | `-port` *`Port_number`*

Specifies the web service port number that is used for communication with VIMServiceHost windows service.

Required?

False

Position

2

Default value

808

Accept pipeline input?

False

Accept wildcard characters?

False

Example: Resuming SnapMirror relationships

The following cmdlet resumes SnapMirror relationships based on information in the disaster recovery plan, *Sunnyvale_Seattle_DRPlan.xml*

```
Invoke-OCDRMirrorResume -DRPlan C:\ProgramData\OnCommand\MS_Plugin
\Sunnyvale_Seattle_DRPlan.xml -Verbose
```

Invoke-OCDRMirrorResync cmdlet

The `Invoke-OCDRMirrorResync` cmdlet resynchronizes SnapMirror relationships based on the information in the disaster recovery plan or the source and destination specified in the cmdlet. This cmdlet reestablishes the SnapMirror configuration from the primary site to the secondary site. The original production site then becomes the active production site again.

Syntax

```
Invoke-OCDRMirrorResync
[[<-DRPlan, drp>] <String> DRPlan_path_with_file_name]
[[<-DestinationLocation, destloc>] <String> Destination_location]
[[<-DestinationSnapshot, destsnap>] <String> Snapshot_destination_location]
[[<-MaxTransferRate, maxrate>] <String> Maximum_transfer_rate_in_KBPS]
[[<-SourceLocation, srcloc>] <String> Source_location]
[[<-SourceSnapshot, srcsnap>] <String> Snapshot_source_location]
[[<-Port>] <String> Port_number]
[[<-CommonParameters>]]
```

Description

This activity requires an optional disaster recovery plan or source and destination locations for executing the operation. If the resynchronization operation fails for any relationship, a Windows event log message that contains the error information is created and the resynchronization operation continues for other SnapMirror relationships.

The update occurs asynchronously, and a schedule is set in `/etc/snapmirror.conf` for the destination. This cmdlet is run on the destination Hyper-V host, and that host must contain the local storage system credentials for both the primary and secondary controllers.

The resynchronization operation also updates the disaster recovery plan with the new primary-to-secondary relationships that are established.

Parameters

`-DRPlan | -drp` *`DRPlan_path_with_file_name`*

Specifies the disaster recovery path by using the disaster recovery plan file name.

Required?

True

Position

0

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-DestinationLocation` | **`-destloc`** ***`Destination_location`***

Specifies the destination of the mirror pair that you want to resynchronize. The destination should be expressed in either of the following forms:

- The volume form:

  `<filer>:<volume>`

- The qtree form:

  `<filer>:/vol/<volume>/<qtree>`

  Both source and destination locations must be provided when the disaster recovery plan is not provided.

Required?

False

Position

1

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-DestinationSnapshot` |**`destsnap`** ***`Snapshot_destination_location`***

Specifies where the Snapshot copy is created after the SnapMirror transfer finishes.

Required?

False

Position

2

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-MaxTransferRate` | **`-maxrate`** ***`Maximum_transfer_rate_in_KBPS`***

Specifies the maximum transfer rate, in kilobytes per second. The default is the highest speed at which the storage system can transfer data.

Required?

False

Position

3

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-SourceLocation` | **`-srcloc`** ***`Source_location`***

Specifies the source of the mirror pair relationship that you want to resynchronize. The source should be expressed in either of the following forms:

- The volume form:

  `<filer>:<volume>`

- The qtree form:

  `<filer>:/vol/<volume>/<qtree>`

  Both source and destination locations must be provided when the disaster recovery plan is not provided.

Required?

False

Position

4

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-SourceSnapshot | -srcsnap` ***`Snapshot_source_location`***

Specifies which source Snapshot copy to use for a qtree update. If the `SourceSnapshot` parameter is not specified, a new Snapshot copy is created.

Required?

False

Position

5

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-Port | -port` ***`Port_number`***

Specifies the web service port number that is used for communication with VIMServiceHost windows service.

Required?

False

Position

6

Default value

808

Accept pipeline input?

False

Accept wildcard characters?

False

Example: Resynchronizing SnapMirror relationships

The following cmdlet resynchronizes the SnapMirror relationships based on the SnapMirror information in the disaster recovery plan *`Sunnyvale_Seattle_DRPlan.xml`*:

```
Invoke-OCDRMirrorResync -DRPlan C:\ProgramData\OnCommand\MS_Plugin
\Sunnyvale_Seattle_DRPlan.xml -Verbose
```

Invoke-OCDRMirrorReverseResync cmdlet

The `Invoke-OCDRMirrorReverseResync` cmdlet reverses the resynchronization process and resynchronizes the mirror relationships from the secondary to the primary storage system based on information in the specified disaster recovery plan. Alternatively, you can specify a source location and a destination location to resynchronize to.

Syntax

```
Invoke-OCDRMirrorReverseResync
[[<-DRPlan, drp>] <String> DRPlan_path_with_file_name]
[[<-DestinationLocation, destloc>] <String> Destination_location]
[[<-SourceLocation, srcloc>] <String> Source_location]
[[<-Port>] <String> Port_number]
[[<-CommonParameters>]]
```

Description

If SnapMirror configurations from the secondary site to the primary site existed prior to the failover, this cmdlet reestablishes the SnapMirror copy configurations after the failover finishes.

If SnapMirror copy configurations from the secondary site to the primary site do not exist prior to the failover, the `Invoke-OCDRMirrorReverseResync` cmdlet creates them. As a result, the volumes on the original primary site become the mirror copies, storing data replicated from the original secondary site.

Before you run the `Invoke-OCDRMirrorReverseResync` cmdlet on Site A, the disaster recovery plan must be in the failback direction, which is achieved by running the `Update-OCDRplan` with the `Failback` parameter on Site B.

The reverse-resynchronization transfer is handled asynchronously; therefore, you must wait for the transfer to finish before you execute any additional operations. The `Invoke-OCDRMirrorReverseResync` cmdlet must be issued on the Hyper-V host.

Parameters

`-DRPlan` | `-drp` *`DRPlan_path_with_file_name`*

Specifies the disaster recovery path, including the disaster recovery plan file name.

Required?

True

Position

0

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-DestinationLocation` | **`-destloc`** ***`Destination_location`***

Specifies the destination location of the mirror relationships that you want to resynchronize in reverse, using one of the following forms:

- The volume form:

  `<filer>:<volume>`

- The qtree form:

  `<filer>:/vol/<volume>/<qtree>`

Required?

False

Position

1

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-SourceLocation` | **`-srcloc`** ***`Source_location`***

Specifies the source location of the update transfer, using the source location expressed in one of the following forms:

- The volume form:

  `<filer>:<volume>`

- The qtree form:

  `<filer>:/vol/<volume>/<qtree>`

  If the source location is not specified, then the source in `/etc/snapmirror.conf` is used for the source path.

Required?

False

Position

1

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-Port` | `-port` *`Port_number`*

Specifies the web service port number that is used for communication with VIMServiceHost windows service.

Required?

False

Position

2

Default value

808

Accept pipeline input?

False

Accept wildcard characters?

False

Example: Reversing the resynchronization process

The following cmdlet reverses the resynchronization process and resynchronizes the mirror relationships from the secondary to the primary storage system based on information in the disaster recovery plan, *`Sunnyvale_Seattle_DRPlan.xml`*:

```
Invoke-OCDRMirrorReverseResync -DRPlan C:\ProgramData\OnCommand
\MS_Plugin\Sunnyvale_Seattle_DRPlan.xml -Verbose
```

Invoke-OCDRMirrorUpdate cmdlet

The `Invoke-OCDRMirrorUpdate` cmdlet updates the mirror relationships on the primary and secondary storage systems using the data captured in the specified disaster recovery plan or from a specified location. The update is asynchronous and must be issued on the destination storage system.

Syntax

```
Invoke-OCDRMirrorUpdate
[[<-DRPlan, drp>] <String> DRPlan_path_with_file_name]
[[<-DestinationLocation, destloc>] <String> Destination_location]
[[<-DestinationSnapshot, destsnap>] <String> Destination_SnapShot]
[[<-MaxTransferRate, maxrate>] <String> KBPS]
[[<-SourceLocation, srcloc>] <String> Source_location]
[[<-SourceSnapshot, srcsnap>] <String> Source_SnapShot]
[[<-Port>] <String> Port_number]
[[<-CommonParameters>]]
```

Description

On Windows 2012 servers, you need the SnapMirror update for restoring virtual machines on secondary sites, when virtual machine configuration files are removed or deleted. However, on Windows 2008/R2 servers, the virtual machines are restored from their replicated backup virtual machine configuration files, which are captured along with the disaster recovery plan.

Parameters

`-DRPlan` | `-drp` *`DRPlan_path_with_file_name`*

Specifies the disaster recovery path by using the disaster recovery plan file name.

Required?

True

Position

0

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-ClearCheckpoint` | `-ccp`

If true, the restart checkpoint is cleared.

Required?

False

Position

1

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-DestinationLocation` | **`-destloc`** ***`Destination_location`***

Specifies the destination of the transfer that you want to update in one of two forms:

- The volume form:

  `<filer>:<volume>`

- The qtree form:

  `<filer>:/vol/<volume>/<qtree>`

Required?

False

Position

2

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-DestinationSnapshot` | **`destsnap`** ***`Destination_snapshot`***

Specifies the destination to which the specified Snapshot copy (in addition to the SnapMirror Snapshot copy) is created after the qtree SnapMirror transfer finishes.

Required?

False

Position

3

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-MaxTransferRate` | **`-maxrate`** ***`Maximum_transfer_rate_in_KBPS`***

Specifies the maximum transfer rate, in kilobytes per second. The default is the highest speed at which the storage system can transfer data.

Required?

False

Position

4

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-SourceLocation` | **`-srcloc`** ***`Source_location`***

Specifies the source of the update transfer in one of two forms:

- The volume form:

  `<filer>:<volume>`

- The qtree form:

  `<filer>:/vol/<volume>/<qtree>`

If the `-SourceLocation` parameter is not specified, then the source in `/etc/snapmirror.conf` is used.

Required?

False

Position

5

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-SourceSnapshot | -srcsnap` *`Source_snapshot`*

Specifies which Snapshot copy to use for a qtree update. If the `SourceSnapshot` parameter is not specified, a new copy is created.

Required?

False

Position

6

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-Port | -port`

Specifies the web service port number that is used for communication with VIMServiceHost windows service.

Required?

False

Position

7

Default value

808

Accept pipeline input?

False

Accept wildcard characters?

False

Example: Updating SnapMirror relationships

The following cmdlet updates the SnapMirror relationships based on information in the disaster recovery plan *`Sunnyvale_Seattle_DRPlan.xml`*:

```
Invoke-OCDRMirrorUpdate -DRPlan "C:\ProgramData\OnCommand\MS_Plugin
\Sunnyvale_Seattle_DRPlan.xml" -Verbose
```

Get-OCDRMirrorStatus cmdlet

The `Get-OCDRMirrorStatus` cmdlet returns the status of all SnapMirror relationships based on the disaster recovery plan. If you specify a source or destination location, the cmdlet returns the SnapMirror status for the specified location. You run this cmdlet on the Hyper-V host, which must contain the credentials for both the primary and secondary controllers in its local storage system.

Syntax

```
Get-OCDRMirrorStatus
[[<-DRPlan, drp>] <String> DRPlan_path_with_file_name]
[[<-Location, loc>] <String> Source_or_destination_location]
[[<-Port>] <String> Port_number]
[[<-CommonParameters>]]
```

Parameters

`-DRPlan | -drp` *`DRPlan_path_with_file_name`*

Specifies the disaster recovery path, including the disaster recovery plan file name.

Required?

True

Position

0

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-Location | -loc` *`Source_or_destination_location`*

Specifies the location of the SnapMirror pair. You must express the location in either of the following forms:

- The volume form:

  `<filer>:<volume>`

- The qtree form:

  `<filer>:/vol/<volume>/<qtree>`

Required?

False

Position

1

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-Port` | **`-port`** ***`Port_number`***

Specifies the web service port number that is used for communication with VIMServiceHost windows service.

Required?

False

Position

2

Default value

808

Accept pipeline input?

False

Accept wildcard characters?

False

Example: Capturing SnapMirror relationship information

The following cmdlet captures the SnapMirror relationships for the storage controller *`PrimarySite`* and the volume *`SourceVolume`*:

```
Get-OCDRMirrorStatus -Location PrimarySite:SourceVolume -Verbose
```

Reset-OCDRSite cmdlet

The `Reset-OCDRSite` cmdlet deletes conflicting cluster resources or disconnects all of the LUNs that are on the secondary site, based on information in the disaster recovery plan.

Syntax

```
Reset-OCDRSite
[[<-DRPlan, drp>] <String> DRPlan_path_with_file_name]
[[<-Force, f>]] <SwitchParameter>]
[[<-Full, fl>]] <SwitchParameter>]
[[<-Port>]] <String> Port_number]
[[<-CommonParameters>]]
```

Description

You must run the `Reset-OCDRSite` cmdlet on the site on which you want to remove conflicts. For example, after a failback, with the disaster recovery plan still in the failback direction, you could run the `Reset-OCDRsite` cmdlet on Site B to remove conflicting virtual machines.

For a normal failover-failback scenario, you should use the `Reset-OCDRSite` cmdlet with the `Force` parameter, to remove conflicting resources on the secondary site. The cmdlet cleans up the following conflicts:

- Conflicting virtual machine cluster resources
- Virtual machine configuration resources
- Dedicated or shared disks with conflicting mount points
- CSV reparse points
- Conflicting virtual machines with the same GUID on the secondary site

When both the `Force` and the `Full` parameters are specified, the `Reset-OCDRSite` cmdlet removes both the conflicting and the non-conflicting resources that are captured in the disaster recovery plan. If virtual machines or resources are not included in the disaster recovery plan, they are not affected.

Parameters

`-DRPlanName | -drp` ***`DRPlan_path_with_file_name`***

Specifies the disaster recovery path, including the disaster recovery plan file name.

Required?

True

Position

0

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-Force |-f`

Specifies that the conflicting LUNs on the secondary site be disconnected or deleted and then conflicting cluster resources on the secondary cluster be removed.

Required?

False

Position

2

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-Full|-fl`

When the `Full` parameter is specified, all cluster shared volumes and all virtual machines are removed from the specified cluster.

The `Full` parameter must be used with extreme caution, because it cleans up all CSVs and virtual machines in the disaster recovery plan or that conflict with the disaster recovery plan.

Required?

False

Position

3

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

-Port | -port

Specifies the web service port number that is used for communication with VIMServiceHost.

Required?

False

Position

4

Default value

808

Accept pipeline input?

False

Accept wildcard characters?

False

Example: Deleting or disconnecting conflicting resources

This example disconnects or deletes all of the conflicting resources on the secondary site based on the disaster recovery plan that resides at the location **`C:\ProgramData\OnCommand\MS_Plugin\SFO_NY_DRPlan.xml`**. After this operation finishes successfully, all the resources (cluster physical disk resources, cluster resource groups, cluster virtual machine resources, cluster virtual machine configuration resources, and disaster recovery LUNs) are deleted or disconnected on the secondary site. The `Full` parameter indicates that all the resources captured in the disaster recovery plan must be deleted on the secondary site. If the `Force` parameter is not specified, then the resources are not deleted or disconnected on the secondary site, based on the disaster recovery plan.

```
Reset-OCDRSite -DRPlanName C:\ProgramData\OnCommand\MS_Plugin
\SFO_NY_DRPlan.xml -Verbose -full -force
```

Restore-OCDRVM cmdlet

The `Restore-OCDRVM` cmdlet restores the virtual machines on the primary and secondary sites based on the information in the specified disaster recovery plan.

Syntax

```
Restore-OCDRVM
[[<-DRPlan, drp>] <String> DRPlan_path_with_file_name]
[[<-OnlineVM, online>] <SwitchParameter>]
[[<-Port>] <String> Port_number]
[[<-CommonParameters>]]
```

Description

This cmdlet restores the virtual machines on the primary and secondary sites based on the information in the specified disaster recovery plan. The virtual machines are restored from the exported virtual machine configuration data that is in the plan.

On the Windows 2008 R2 and Windows 2012 platforms, you can recover your Hyper-V virtual machines that are online on your primary site and bring them online on your secondary sites. You can restore your standalone or HA virtual machines that are online on your primary site onto secondary sites with one click and very limited down-time.

There are two important differences between Windows 2008 R2 and Windows 2012.

With Windows 2012, you can create and update your disaster recovery plan, which is required to perform failover and failback, by running the cmdlets `New-OCDRplan` and `Update-OCDRplan` while the virtual machines are online.

With Windows 2008 R2, all the virtual machines have to be turned off or shutdown to update the disaster recovery plan. For Windows 2012 servers, the virtual machine configuration files are not generated, however on Windows 2008 R2 servers, virtual machine configuration files are generated along with the disaster recovery plan file.

Parameters

`-DRPlan | -drp` ***`DRPlan_path_with_file_name`***

Specifies the disaster recovery path, including the disaster recovery plan file name.

Required?

True

Position

0

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-OnlineVM | -online`

Brings the virtual machines online after they are restored.

Required?

False

Position

1

Default value

Off

Accept pipeline input?

False

Accept wildcard characters?

False

`-Port` | `-port` ***`Port_number`***

Specifies the web service port number that is used for communication with VIMServiceHost windows service.

Required?

False

Position

2

Default value

808

Accept pipeline input?

False

Accept wildcard characters?

False

Example: Restoring virtual machines on the primary and secondary sites

The following cmdlet restores the virtual machines on the primary and secondary sites based on the information in disaster recovery plan, *`Sunnyvale_Seattle_DRPlan.xml`*. By default all the virtual machines are in an **`Off`** state.

```
Restore-OCDRVM -DRPlan C:\ProgramData\OnCommand\MS_Plugin
\Sunnyvale_Seattle_DRPlan.xml -Verbose
```

Orchestrator disaster recovery activities

You use Orchestrator disaster recovery activities to create workflows that perform failover when your system experiences a failure. You also use these activities to create and manage a disaster recovery plan and maintain your SnapMirror relationships.

Create DR Plan activity

You use the **Create DR Plan** activity to specify the information for the disaster recovery plan, which is used in the event of a failover of your system.

Description

The disaster recovery plan describes disaster recovery activities and determines how they behave during failover and failback. The plan is stored in a user-specified location as an XML file. The default folder location is `C:\ProgramData\OnCommand\MS_Plugin`.

You create the plan on the primary site, which captures the information about the primary setup so that it can be used to replicate the configuration on the secondary site upon failover. The plan captures only the information that is required to perform failover and failback, including the following information:

- A list of the virtual machines on the primary site and the detailed virtual machine properties
- Storage information for the configuration of virtual machines, VHDs, and Snapshot data
- SnapMirror relationships for all volumes on the primary storage system on which the virtual machines are running
- The primary and secondary host-related or cluster-related information

Properties

The following table indicates the information that you can enter in the properties tab for the **Create DR Plan** activity:

Property name	**Type**	**Required?**	**Default**
DR plan name [1]	String	Required	None
DR plan in folder [2]	String	Required	`C:\ProgramData\OnCommand\MS_Plugin`
Primary server name [3]	String	Required	None
Secondary server name [4]	String	Required	None
Operation direction [5]	Choice	Optional	Failover

1. Indicates the name of the disaster recovery plan
2. Indicates either a local path or a CIFS path
   The VIM web service must have write access on this folder.
3. Indicates the primary host name or cluster node name. The XML plan file is created on the primary server.

4. Indicates the secondary host name or cluster node name
5. Indicates the direction of the current operation: either failover or failback

Published data

The following table indicates the information that the plug-in can return from the **Create DR Plan** activity:

Name	Type
DR plan path [1]	String

1. Indicates the full path, including the plan name, to the disaster recovery plan

Validate DR Plan activity

You use the **Validate DR Plan** activity to validate the current state of the secondary storage system against the data captured in the specified disaster recovery plan whenever the primary or secondary site configuration changes.

Description

The **Validate DR Plan** activity checks the following factors:

- Ensures that the secondary host is running on a Windows 2008 R2 or later operating system and that the host type is the same as the live host
- Ensures that the virtual machine LUNs that are to be connected on the secondary site have valid LUN paths and SnapMirror relationships
- Verifies the active SnapMirror status and the SnapMirror relationships and ensures that all the SnapMirror relationships are in the `snapmirrored` state
- Verifies that the virtual machine cluster and configuration resources are the same as the live virtual machine
- Ensures that there is no conflict with the cluster and configuration resources on the secondary host

  If a conflict is found in the mount points, the **Validate DR Plan** activity checks the mount point that is connected in the secondary host to determine if it is connected to the required LUN on the secondary storage system.
- Verifies that the storage system credentials are configured for all the storage systems in the disaster recovery process on the secondary site
- Ensures that the Hyper-V role is enabled on the secondary host or cluster

- If the disaster recovery plan is created in the context of a cluster, ensures that the cluster shared volume is active on the secondary cluster and that a valid cluster name and IP address resources are present and online

Properties

The following table indicates the information that you can enter in the properties tab for the **Validate DR Plan** activity:

Property name	Type	Required?	Default
Primary server name [1]	String	Required	None
DR plan path [2]	String	Required	None

1. Indicates the primary host name or cluster node name. The XML plan file is created on the primary server.
2. Indicates the full path, including the plan name, to the disaster recovery plan

Published data

The following table indicates the information that the plug-in can return from the **Validate DR Plan** activity:

Name	Type
DR plan path [1]	String

1. Indicates the full path, including the plan name, to the disaster recovery plan

Update a DR Plan activity

You use the **Update DR Plan** activity to update the disaster recovery plan with information from both the host and storage.

Properties

The following table indicates the information that you can enter in the properties tab for the **Update DR Plan** activity:

Property name	Type	Required?	Default
DR plan path	String	Required	None
Primary server name [1]	String	Required	None
Secondary server name [2]	String	Required	None
Operation direction [3]	Choice	Optional	Failover

1. Indicates the Hyper-V primary server name.
2. Indicates the Hyper-V secondary server name.
3. Indicates the direction of the operation: either failover or failback.

Published data

The following table indicates the information that the plug-in can return **Update DR Plan** activity:

Name	Type
DR plan path [1]	String

1. Indicates the full path, including the plan name, to the disaster recovery plan

Connect DR Storage activity

You use the **Connect DR Storage** activity to connect to the storage on the specified host or cluster, based on the disaster recovery plan.

Properties

The following table provides the information that you can enter in the properties tab for the **Connect DR Storage** activity:

Property name	Type	Required?	Default
DR plan path [1]	String	Required	None
Secondary server name	String	Required	None
Force disconnect conflicted storage before connect	Boolean	Optional	False

1. Indicates the full path, including the plan name, to the disaster recovery plan

Published data

The following table provides the information that the plug-in can return for the **Connect DR Storage** activity:

Name	Type
CSV owner node name	String
Device ID	String
Disk cluster resource group name	String
Disk cluster resource name	String

Name	**Type**
Disk resource name	String
Disk resource owner node name	String
Disk serial number	String
Is CSV disk [1]	Boolean
Is cluster disk [2]	Boolean
Mount point	String
iSCSI address	String
Storage path	String
Storage size [3]	String
Volume GUID	String

1. Returns `True` if the storage is a cluster shared volume; otherwise, it returns `False`
2. Returns `True` if the storage is a cluster disk; otherwise, it returns `False`
3. Returns the storage size, displayed in bytes.

Clean-up a DR site activity

You use the **Clean-up a DR Site** activity to delete or disconnect conflicting cluster resources or LUNs that are on the current secondary site, based on information in the disaster recovery plan.

You can specify the `Secondary server name` property to remove conflicting resources on the secondary site before the failback.

For a normal failover or failback scenario, you should use the **Clean-up a DR Site** activity without any properties, to remove conflicting resources on the secondary site.

The activity removes the following conflicts:

- Conflicting virtual machine cluster resources
- Virtual machine configuration resources
- Dedicated or shared disks with conflicting mount points
- CSV reparse points
- Conflicting virtual machines with the same GUID on the secondary site

Properties

The following table indicates the information that you can enter in the properties tab for the **Clean-up a DR Site** activity:

Property name	Type	Required?	Default
DR plan path [1]	String	Required	None
Secondary server name [2]	String	Required	None
Remove DRPlan resources and conflicting LUNs [3]	Boolean	Required	False
Remove DRPlan resources, LUNs and conflicting LUNs	Boolean	Optional	False
Primary server name	String	Optional	None

1. Indicates the full path, including the plan name, to the disaster recovery plan.
2. Indicates the secondary host that has the VIM web service installed.
3. Ensures that you want to run the **Clean-up a DR Site** object.

Published data

The following table indicates the information that the plug-in can return from the **Clean-up a DR Site** activity:

Name	Type
DR plan path [1]	String

1. Indicates the full path, including the plan name, to the disaster recovery plan

Restore VMs activity

You use the **Restore VMs** activity to restore those virtual machines that are specified in the disaster recovery plan on the primary and secondary sites.

Properties

The following table indicates the information that you can enter in the properties tab for the **Restore VMs** activity:

Property name	Type	Required?	Default
DR plan path [1]	String	Required	None
VM name	String	Optional	None

Property name	Type	Required?	Default
Secondary server name [2]	String	Required	None
Online VMs after restore	Boolean	Optional	False

1. Indicates the full path, including the plan name, to the disaster recovery plan
2. Indicates the secondary host that has VIM web service installed.

Published data

The following table indicates the information that the plug-in can return from the **Restore VMs** activity:

Name	Type
Result details	String
Result message	String

Abort SnapMirror activity

You use the **Abort SnapMirror** activity to cancel data transfer to a specified destination, based on information in the disaster recovery plan.

Properties

The following table indicates the information that you can enter in the properties tab for the **Abort SnapMirror** activity:

Property name	Type	Required?	Default, if any
Destination path [1]	String	Required	None
Secondary server name [2]	String	Required	None
Clear restart checkpoint [3]	True/False	Optional	False
DR plan path [4]	String	Optional	None

1. Indicates the name or IP address of the controller.
2. Indicates where the VIM web service has been installed
3. Indicates whether to clear the SnapMirror restart checkpoint.
4. Indicates the full path, including the plan name, to the disaster recovery plan

Published data

The following able indicates the output that you can include in the **Abort SnapMirror** activity:

Name	Type
Results details	String
Result message	String
SnapMirror result	String

Break SnapMirror activity

You use the **Break SnapMirror** activity to break all of the SnapMirror relationships on the secondary storage system, based on information in the disaster recovery plan. If no disaster recovery plan path is specified, the **Break SnapMirror** activity breaks a SnapMirror relationship based on the `Destination Path` property.

Properties

The following table indicates the information that you can enter in the properties tab for the **Break SnapMirror** activity:

Property name	Type	Required?	Default
Destination path [1]	String	Required	None
Secondary server name [2]	String	Required	None
DR plan path [3]	String	Optional	None

1. Indicates the SnapMirror relationship by providing the destination path of the relationship.
2. Indicates where the VIM web service has been installed
3. Indicates the full path, including the plan name, to the disaster recovery plan

Published data

The following table indicates the output that is available from the **Break SnapMirror** activity:

Name	Type
Results details	String
Result message	String
SnapMirror result	String

Get SnapMirror Status activity

You use the **Get SnapMirror Status** activity to return the status of all SnapMirror relationships, based on information in the disaster recovery plan. If you specify a source or destination location, the SnapMirror status for the specified location is returned.

Properties

The following table indicates the information that you can enter in the properties tab for the **Get SnapMirror Status** activity:

Property	Type	Required?	Default
Destination path [1]	String	Required	None
Secondary server name [2]	String	Required	None
DR plan path [3]	String	Optional	None

1. Indicates the controller name or IP address
2. Indicates where the VIM web service is installed
3. Indicates the full path, including the plan name, to the disaster recovery plan

Published data

The following table indicates the output that is available from the **Get SnapMirror Status** activity:

Name	Type
Base Snapshot [1]	String
Contents [2]	String
Current transfer error [3]	String
Current transfer type [4]	String
Destination path	String
Lag time	String
Last transfer duration [5]	String
Last transfer from [6]	String
Last transfer size [7]	String
Last transfer type [8]	String
Mirror timestamp [9]	String

Name	Type
Results details	String
Result message	String
Source path	String
State [10]	String
Status [11]	String
Transfer progress [12]	String

1. Indicates the base Snapshot name
2. Indicates the state of the active file system of SnapMirror destinations. Possible values are **`replica`**, **`transitioning`**, and **`original`**.
3. Indicates an error during the current SnapMirror transfer
4. Indicates the type of the current SnapMirror transfer, if any. Possible values are **`initialize`**, **`store`**, **`schedule`**, **`retry`**, **`retrieve`**, **`resync`**, and **`migrate`**.
5. Indicates the duration, in seconds, of the most recent SnapMirror transfer
6. Indicates the source location of the most recent SnapMirror transfer
7. Indicates the size, in KB, of the most recent transfer
8. Indicates the previous SnapMirror transfer type, if any. Possible values are **`initialize`**, **`store`**, **`schedule`**, **`retry`**, **`retrieve`**, **`resync`**, and **`migrate`**.
9. Indicates the access time, in seconds, since January 1, 1970
10. Indicates possible states, including:
    - Broken-off
    - Quiesced
    - Snapmirrored
    - Source
    - Uninitialized
    - Unknown
11. Indicates possible states, including:
    - Aborting

- Idle
- Idle with restart checkpoint
- Pending
- Pending with restart checkpoint
- Transferring

12. Indicates the number of kilobytes transferred during the current transfer

Initialize SnapMirror activity

You use the **Initialize SnapMirror** activity to begin a complete transfer of information from the primary site to the secondary site to establish a SnapMirror relationship for the first time. This process is known as *initializing a destination*.

Properties

The following table indicates the information that you can enter in the properties tab for the **Initialize SnapMirror** activity:

Property name	Type	Required?	Default
Source path	String	Required	None
Destination path	String	Required	None
Secondary server name [1]	String	Required	None
Destination Snapshot name	String	Optional	None
Max transfer speed [2]	String	Optional	None
Source Snapshot name	String	Optional	None

1. Indicates where the VIM web service has been installed
2. Indicates the maximum transfer speed, in KB, per second

Published data

The following table indicates the output that you can include in the **Initialize SnapMirror** activity:

Name	Type
Results details	String
Result message	String
SnapMirror result	String

Quiesce SnapMirror activity

You use the **Quiesce SnapMirror** activity to temporarily quiesce the mirror relationships based on information in the specified disaster recovery plan. You must check the status of the mirror relationships to be sure that they have quiesced. All scheduled SnapMirror transfers are halted until the quiesce process finishes.

Properties

The following table indicates the information that you can enter in the properties tab for the **Quiesce SnapMirror** activity:

Property name	**Type**	**Required?**	**Default**
Destination path [1]	String	Required	None
Secondary server name [2]	String	Required	None
DR plan path [3]	String	Optional	None

1. Indicates the controller name or IP address
2. Indicates where the VIM web service has been installed
3. Indicates the full path, including the plan name, to the disaster recovery plan

Published data

The following table indicates the output that you can include in the **Quiesce SnapMirror** activity:

Name	**Type**
Results details	String
Result message	String
SnapMirror result	String

Release SnapMirror activity

You use the **Release SnapMirror** activity on the primary site to release a SnapMirror relationship and to indicate that it no longer requests updates. The **Release SnapMirror** activity works based on information in the disaster recovery plan and applies to the secondary storage system.

Properties

The following table indicates the information that you can enter in the properties tab for the **Release SnapMirror** activity:

Property name	**Type**	**Required?**	**Default**
Source path	String	Required	None
Destination path	String	Required	None
Primary server name [1]	String	Required	None

1. Indicates where the VIM web service has been installed

Published data

The following table indicates the output that is available from the **Release SnapMirror** activity:

Name	**Type**
Results details	String
Result message	String
SnapMirror result	String

Resume SnapMirror activity

You use the **Resume SnapMirror** activity to restore the mirror relationships based on information in the specified disaster recovery plan. Transfers that were previously paused (quiesced) resume to the specified destination.

Properties

The following table indicates the information that you can enter in the properties tab for the **Resume SnapMirror** activity:

Property name	**Type**	**Required?**	**Default**
Destination path	String	Required	None
Secondary server name [1]	String	Required	None
DR plan path [2]	String	Optional	None

1. Indicates where the VIM web service has been installed
2. Indicates the full path, including the plan name, to the disaster recovery plan

Published data

The following table indicates the output that is available from the **Resume SnapMirror** activity:

Name	Type
Results details	String
Result message	String
SnapMirror result	String

Resync SnapMirror activity

You use the **Resync SnapMirror** activity to resynchronize the SnapMirror relationships, based on the information in the disaster recovery plan. This activity reestablishes the SnapMirror configurations from the primary site to the secondary site after the failback finishes. The original production site then becomes the active production site again. This activity requires a disaster recovery plan and destination.

Description

If the resynchronization operation fails for any relationship, a Windows event log message that contains the error information is created and the resynchronization operation continues for other SnapMirror relationships.

The update occurs asynchronously and a schedule is set in `/etc/snapmirror.conf` for the destination. The resynchronization operation updates the plan with the new primary-to-secondary relationships that are established.

Properties

The following table indicates the information that you can enter in the properties tab for the **Resync SnapMirror** activity:

Property name	Type	Required?	Default
Destination path	String	Required	None
Secondary server name [1]	String	Required	None
Source path	String	Optional	None
DR plan path[2]	String	Optional	None

1. Indicates where the VIM web service has been installed
2. Indicates the full path, including the plan name, to the disaster recovery plan

Published data

The following table indicates the output that is available from the **Resync SnapMirror** activity:

Name	Type
Results details	String
Result message	String
SnapMirror result	String

Reverse Resync SnapMirror activity

You use the **Reverse Resync SnapMirror** activity to reverse the resynchronization process and resynchronize the mirror relationships from the secondary to the primary storage system, based on information in the specified disaster recovery plan. Alternatively, you can specify a destination location to resynchronize to.

Description

If SnapMirror configurations from the secondary site to the primary site existed prior to the failover, this object reestablishes the SnapMirror copy configurations after the failover finishes. If SnapMirror copy configurations from the secondary site to the primary site do not exist prior to the failover, this activity creates them. The original production site then becomes the active production site again.

This activity requires that you specify either a destination location for executing the resynchronization operation or a disaster recovery plan, which contains destination location information. The reverse resynchronization transfer is handled asynchronously; therefore, you must wait for the transfer to finish before you execute any additional operations.

Properties

The following table indicates the information that you can enter in the properties tab for the **Reverse Resync SnapMirror** activity:

Property name	Type	Required?	Default
Destination path	String	Required	None
Secondary server name [1]	String	Required	None
Source path	String	Optional	None
DR plan path [2]	String	Optional	None

1. Indicates where the VIM web service has been installed
2. Indicates the full path, including the plan name, to the disaster recovery plan

Published data

The following table indicates the output that is available from the **Reverse Resync SnapMirror** activity:

Name	Type
Results details	String
Result message	String
SnapMirror result	String

Update SnapMirror activity

You use the **Update SnapMirror** activity to update the mirrors on the destination with the contents of the source. The update is asynchronous and must be issued on the destination storage system.

Properties

The following table indicates the information that you can enter in the properties tab for the **Update SnapMirror** activity:

Property name	Type	Required?	Default
Secondary server name [1]	String	Required	None
DRPlan [2]	String	Optional	None
Source path [3]	String	Optional	None
Destination path [4]	String	Optional	None
Destination Snapshot name	String	Optional	None
Max transfer speed [5]	String	Optional	None
Source Snapshot name	String	Optional	None

1. Indicates where the VIM web service has been installed.

   The required parameters are the secondary server name and either the destination path or the DRPlan.

2. You cannot specify both the disaster recovery plan name and the destination path.

   You must specify a destination path if no disaster recovery path is specified.

3. You do not need to the specify source path if it is listed in the `snapmirror.conf` file.

4. You cannot specify both the disaster recovery plan name and the destination path.

   You must specify a destination path if no disaster recovery path is specified.

5. Indicates the maximum transfer speed, in KB per second.

Published data

The following table indicates the output that is available from the **Update SnapMirror** activity:

Name	Type
Results details	String
Result message	String
SnapMirror result	String

Disaster recovery composite cmdlets

You can use disaster recovery cmdlets to control failover and failback procedures, make changes to the disaster recovery plan, and control the SnapMirror relationships. Composite cmdlets are created from a group of individual cmdlets that are performed together, in a preset order.

Invoke-OCDRFailover cmdlet

The `Invoke-OCDRFailover` cmdlet is a composite operation that calls several individual cmdlets to automate the failover of virtual machines from a primary site to a secondary site.

Syntax

```
Invoke-OCDRFailover
[[<-DRPlan, drp>] <String>DRPlan_path_with_file_name]
[[<-OnlineVM, online>] <SwitchParameter>]
[[<-Force, f>] <SwitchParameter>]
[[<-Port>] <String> Port_number]
[[<-CommonParameters>]]
```

Description

This is a composite cmdlet that calls the individual cmdlets to automate the failover of virtual machines from a primary site to a secondary site. This composite cmdlet performs the following individual cmdlets in order:

1. `Confirm-OCDRPlan`
2. `Reset-OCDRSite`
3. `Invoke-OCDRMirrorBreak`
4. `Connect-OCDRLUN`
5. `Restore-OCDRVM`

On the Windows 2008 R2 and Windows 2012 platforms, you can recover your Hyper-V virtual machines that are online on your primary site and bring them online on your secondary sites. You can restore your standalone or HA virtual machines that are online on your primary site onto secondary sites with one click and very limited down-time.

There are two important differences between Windows 2008 R2 and Windows 2012.

With Windows 2012, you can create and update your disaster recovery plan, which is required to perform failover and failback, by running the cmdlets `New-OCDRplan` and `Update-OCDRplan` while the virtual machines are online.

With Windows 2008 R2, all the virtual machines have to be turned off or shutdown to update the disaster recovery plan. For Windows 2012 servers, the virtual machine configuration files are not generated, however on Windows 2008 R2 servers, virtual machine configuration files are generated along with the disaster recovery plan file.

Parameters

`-DRPlan` | **`-drp`** ***`DRPlan_path_with_file_name`***

Specifies the disaster recovery path, including the disaster recovery plan file name.

Required?

True

Position

0

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-OnlineVM` | **`online`**

Brings the virtual machine online after it is restored.

Required?

False

Position

1

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-Force | -f`

Specifies whether to forcefully disconnect the file system when the LUN is in use, has virtual machines, or has open files. If you do not use this parameter, the file system is not unmounted and the LUN is not disconnected.

Required?

False

Position

2

Default value

False

Accept pipeline input?

False

Accept wildcard characters?

False

`-Port | -port` ***`Port_number`***

Specifies the web service port number that is used for communication with VIMServiceHost windows service.

Required?

False

Position

3

Default value

808

Accept pipeline input?

False

Accept wildcard characters?

False

Example: Performing a disaster recovery failover

The following cmdlet performs a disaster recovery failover to the secondary site based on information in the disaster recovery plan that is located at *`C:\ProgramData\OnCommand\MS_Plugin\SFO_NY_DRPlan.xml`*. If the `Force` parameter is not specified, the conflicting resources are not resolved on the secondary site.

```
Invoke-OCDRFailover -DRPlan C:\ProgramData\OnCommand\MS_Plugin
\SFO_NY_DRPlan.xml -Verbose -Force
```

Related references

Confirm-OCDRPlan cmdlet on page 104
Reset-OCDRSite cmdlet on page 135
Restore-OCDRVM cmdlet on page 137
Invoke-OCDRMirrorBreak cmdlet on page 116
Connect-OCDRLUN cmdlet on page 108

Invoke-OCDRFailback cmdlet

The `Invoke-OCDRFailback` cmdlet is a composite operation that calls individual cmdlets to automate the failback of the virtual machines from a secondary site to a primary site. You must run the `Invoke-OCDRFailback` cmdlet on the secondary site.

Syntax

```
Invoke-OCDRFailback
[[<-DRPlan, drp>] <String>DRPlan_path_with_file_name]
[[<-OnlineVM, online>] <SwitchParameter>]
[[<-Force, f>] <SwitchParameter>]
[[<-Port>] <String> Port_number]
[[<-CommonParameters>]]
```

Description

This cmdlet is a composite activity that calls individual cmdlets to automate the failback of the virtual machines from a secondary site to a primary site. It performs the following individual cmdlets in order:

1. `Confirm-OCDRPlan`
2. `Reset-OCDRSite`
3. `Invoke-OCDRMirrorBreak`
4. `Connect-OCDRLUN`
5. `Restore-OCDRVM`
6. `Invoke-OCDRMirrorReverseResync`

On the Windows 2008 R2 and Windows 2012 platforms, you can recover your Hyper-V virtual machines that are online on your primary site and bring them online on your secondary sites. You can restore your standalone or HA virtual machines that are online on your primary site onto secondary sites with one click and very limited down-time.

There are two important differences between Windows 2008 R2 and Windows 2012.

With Windows 2012, you can create and update your disaster recovery plan, which is required to perform failover and failback, by running the cmdlets `New-OCDRplan` and `Update-OCDRplan` while the virtual machines are online.

With Windows 2008 R2, all the virtual machines have to be turned off or shutdown to update the disaster recovery plan. For Windows 2012 servers, the virtual machine configuration files are not generated, however on Windows 2008 R2 servers, virtual machine configuration files are generated along with the disaster recovery plan file.

Parameters

`-DRPlan | -drp` ***`DRPlan_path_with_file_name`***

Specifies the disaster recovery path, including the disaster recovery path file name.

Required?

True

Position

0

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-OnlineVM | -online`

Brings the virtual machine online after it is restored.

Required?

False

Position

1

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-Force | -f`

Specifies whether to forcefully disconnect the LUN when it is in use, has virtual machines, or has open files. If you do not use this parameter, the file system is not unmounted and the LUN is not disconnected.

Required?

False

Position

2

Default value

False

Accept pipeline input?

False

Accept wildcard characters?

False

`-Port` | **`-port`** ***`Port_number`***

Specifies the web service port number that is used for communication with VIMServiceHost windows service.

Required?

False

Position

3

Default value

808

Accept pipeline input?

False

Accept wildcard characters?

False

Example: Performing a disaster recovery failback

The following cmdlet performs a disaster recovery failback to the primary site based on information in the disaster recovery plan that is located at *`C:\ProgramData\OnCommand\MS_Plugin\SFO_NY_DRPlan.xml`*:

```
Invoke-OCDRFailback -DRPlan C:\ProgramData\OnCommand\MS_Plugin
\SFO_NY_DRPlan.xml -Verbose -Force
```

Related references

Confirm-OCDRPlan cmdlet on page 104
Reset-OCDRSite cmdlet on page 135
Restore-OCDRVM cmdlet on page 137
Invoke-OCDRMirrorBreak cmdlet on page 116
Connect-OCDRLUN cmdlet on page 108

Start-OCDRSimulateFailover cmdlet

The `Start-OCDRSimulateFailover` cmdlet is a composite cmdlet that runs individual operations that simulate, in test mode, failover to the secondary site and that restore all primary virtual machines. You must run the `Start-OCDRSimulateFailover` cmdlet on the secondary site.

Syntax

```
Start-OCDRSimulateFailover
[[<-DRPlan, drp>] <String>DRPlan_path_with_file_name]
[[<-OnlineVM, online>] <SwitchParameter>]
[[<-Force, f>] <SwitchParameter>]
[[<-Port>] <String> Port_number]
[[<-CommonParameters>]]
```

Description

This cmdlet performs the following individual cmdlets on the secondary site, in order:

1. `Confirm-OCDRPlan`
2. `Reset-OCDRSite` (The default is to remove conflicts on the secondary site.)
3. `Invoke-OCDRMirrorBreak`
4. `Connect-OCDRLUN`
5. `Restore-OCDRVM`

On the Windows 2008 R2 and Windows 2012 platforms, you can recover your Hyper-V virtual machines that are online on your primary site and bring them online on your secondary sites. You can restore your standalone or HA virtual machines that are online on your primary site onto secondary sites with one click and very limited downtime.

There are two important differences between Windows 2008 R2 and Windows 2012:

- With Windows 2012, you can create and update your disaster recovery plan, which is required to perform failover and failback, by running the cmdlets `New-OCDRplan` and `Update-OCDRplan` while the virtual machines are online.

- With Windows 2008 R2, all the virtual machines have to be turned off or shut down to update the disaster recovery plan. For Windows 2012 servers, the virtual machine configuration files are not generated; however, on Windows 2008 R2 servers, virtual machine configuration files are generated along with the disaster recovery plan file.

Parameters

`-DRPlan` | **`-drp`** ***`DRPlan_path_with_file_name`***

Specifies the disaster recovery path by using the disaster recovery plan name.

Required?

True

Position

0

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-OnlineVM` | **`-online`**

Brings the virtual machine online after it is restored.

Required?

False

Position

1

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-Force` | **`-f`**

Specifies whether to forcefully disconnect the LUN when it is in use, has virtual machines, or has open files. If you do not use this parameter, the file system is not unmounted and the LUN is not disconnected.

Required?

False

Position

2

Default value

False

Accept pipeline input?

False

Accept wildcard characters?

False

`-Port | -port` ***`Port_number`***

Specifies the web service port number that is used for communication with VIMServiceHost.

Required?

False

Position

3

Default value

808

Accept pipeline input?

False

Accept wildcard characters?

False

Example: Starting the simulated failover

The following cmdlet performs a simulated disaster recovery failover to a secondary site based on the disaster recovery plan *`C:\ProgramData\OnCommand\MS_Plugin\SFO_NY_DRPlan.xml`*:

```
Start-OCDRSimulateFailover -DRPlan C:\ProgramData\OnCommand\MS_Plugin
\SFO_NY_DRPlan.xml -Verbose
```

Related references

Confirm-OCDRPlan cmdlet on page 104
Reset-OCDRSite cmdlet on page 135
Restore-OCDRVM cmdlet on page 137
Invoke-OCDRMirrorBreak cmdlet on page 116
Connect-OCDRLUN cmdlet on page 108

Stop-OCDRSimulateFailover cmdlet

The `Stop-OCDRSimulateFailover` cmdlet is a composite activity that ends the failover simulation and reestablishes the SnapMirror relationship from the primary site to the secondary site.

Syntax

```
Stop-OCDRSimulateFailover
[[<-DRPlan, drp>] <String>DRPlan_path_with_file_name]
[[<-Force, f>] <SwitchParameter>]
[[<-Port>] <String> Port_number]
[[<-CommonParameters>]]
```

Description

This cmdlet resolves conflicts on the secondary site in its entirety. After the conflicts are resolved on the secondary site, the `Invoke-OCDRMirrorResync` cmdlet establishes SnapMirror replication relationships from the primary site to the secondary site to bring the system back to the original state. This cmdlet performs the following individual cmdlets in order:

1. `Reset-OCDRSite` (The default is to resolve conflicts on the secondary site.)
2. `Invoke-OCDRMirrorResync`

Parameters

`-DRPlan` | **`-drp`** ***`DRPlan_path_with_file_name`***

Specifies the disaster recovery path, including the disaster recovery plan file name.

Required?

True

Position

0

Default value

None

Accept pipeline input?

False

Accept wildcard characters?

False

`-Force` | **`-f`**

Specifies whether to forcefully disconnect the LUN when it is in use, has virtual machines, or has open files. If you do not use this parameter, the file system is not unmounted and the LUN is not disconnected.

Required?

False

Position

1

Default value

False

Accept pipeline input?

False

Accept wildcard characters?

False

`-Port | -port` ***`Port_number`***

Specifies the web service port number that is used for communication with VIMServiceHost windows service.

Required?

False

Position

2

Default value

808

Accept pipeline input?

False

Accept wildcard characters?

False

Example: Stopping a simulated failover

The following cmdlet stops a currently running failover simulation to the secondary site based on information in the disaster recovery plan, *`C:\ProgramData\OnCommand\MS_Plugin\SFO_NY_DRPlan.xml`*. All the SnapMirror relationships that are broken as a part of the simulate failover procedure are brought back to the mirrored state.

```
Stop-OCDRSimulateFailover -DRPlan C:\ProgramData\OnCommand\MS_Plugin
\SFO_NY_DRPlan.xml-Verbose I
```

Related references

Reset-OCDRSite cmdlet on page 135

Invoke-OCDRMirrorReverseResync cmdlet on page 126

Automated workflow activities for Orchestrator disaster recovery

You use the automated workflow activities to call several individual activities to automate a workflow. There are automated workflow activities for failover and failback workflows and for the simulated failover workflow.

DR Failover activity

You use the **DR Failover** workflow activity (which is a composite activity comprising five other activities) to automate the failover of virtual machines from a primary site to a secondary site. The component activities are performed in a specific order. The primary and secondary sites change, based on the direction of the workflow.

Description

The **DR Failover** activity is an automated workflow (composite) activity that performs the following individual activities in order:

1. **Validate DR Plan**
2. **Clean-up a DR Site**
3. **Break SnapMirror**
4. **Connect DR Storage**
5. **Restore VMs**

Properties

The following table indicates the information that you can enter in the properties tab for the **DR Failover** activity:

Property name	Type	Required?	Default
DR plan path [1]	String	Required	None
Secondary server name [2]	String	Required	None
Force disconnect conflicted storage before connect	Boolean	Optional	False
Online VMs after restore	Boolean	Optional	False

1. Indicates the full path, including the plan name, to the disaster recovery plan

2. Indicates where the VIM web service has been installed

Published data

The following table indicates the output that is available from the **DR Failover** activity:

Name	Type
Results details	String
Result message	String

DR Failback activity

You use the **DR Failback** workflow activity to automate the failback of virtual machines from a secondary site to a primary site. DR Failback is a composite activity comprising five component activities that are performed in a specific order.

Description

The **DR Failback** activity is an automated workflow (composite) activity that performs the following individual activities in order:

1. **Validate DR Plan**
2. **Clean-up a DR Site**
3. **Break SnapMirror**
4. **Connect DR Storage**
5. **Restore VMs**

Properties

The following table indicates the information that you can enter in the properties tab for the **DR Failback** activity:

Property name	Type	Required?	Default
Secondary server name [1]	String	Required	None
DR plan path [2]	String	Required	None
Force disconnect conflicted storage before connect	Boolean	Optional	False
Online VMs after restore	Boolean	Optional	False

1. Indicates where the VIM web service has been installed

2. Indicates the full path, including the plan name, to the disaster recovery plan

Published data

The following table indicates the output that is available from the **DR Failback** activity:

Name	Type
Results details	String
Result message	String

Start Test DR Failover activity

You use the **Start Test DR Failover** workflow activity to simulate, in test mode, a failover operation to the secondary site and a restoration of all primary virtual machines.

Description

The **Start Test DR Failover** activity is an automated workflow (composite) activity that performs the following individual activities, in order, in the test environment:

1. **Validate DR Plan**
2. **Clean-up a DR Site**
3. **Break SnapMirror**
4. **Connect DR Storage**
5. **Restore VMs**

Properties

The following table indicates the information that you can enter in the properties tab for the **Start Test DR Failover** activity:

Property name	Type	Required?	Default
DR plan path [1]	String	Required	None
Secondary server name [2]	String	Required	None
Force disconnect conflicted storage before connect	Boolean	Optional	False
Online VMs after restore	Boolean	Optional	False

1. Indicates the full path, including the plan name, to the disaster recovery plan
2. Indicates where the VIM web service has been installed

Published data

The following table indicates the output that is available from the **Start Test DR Failover** activity:

Name	Type
Results details	String
Result message	String

End Test DR Failover activity

You use the **End Test DR Failover** automated workflow (composite) activity to end the failover simulation and reestablish the SnapMirror relationship from the primary to the secondary site. This activity also simulates shutting down and deleting the virtual machines or taking their cluster resources offline on the secondary site.

Description

The **End Test DR Failover** activity simulates removing conflicts on the secondary site, followed by the **Resync SnapMirror** activity to establish SnapMirror replication relationships from the primary site to the secondary site to bring the system back to the original state.

This is an automated workflow (composite) activity that performs the following individual activities in the test environment:

1. **Clean-up a DR Site**
2. **Resync SnapMirror**

Properties

The following table indicates the information that you can enter in the properties tab for the **End Test DR Failover** activity:

Property name	Type	Required?	Default
DR plan path [1]	String	Required	None
Secondary server name [2]	String	Required	None
Force disconnect conflicted storage before connect	Boolean	Optional	False

1. Indicates the full path, including the plan name, to the disaster recovery plan
2. Indicates where the VIM web service has been installed

Published data

The following table indicates the output that is available from the **End Test DR Failover** activity:

Name	Type
Results details	String
Result message	String

Copyright information

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.

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.277-7103 (October 1988) and FAR 52-227-19 (June 1987).

Trademark information

NetApp, the NetApp logo, Go Further, Faster, ASUP, AutoSupport, Campaign Express, Cloud ONTAP, clustered Data ONTAP, Customer Fitness, Data ONTAP, DataMotion, Fitness, Flash Accel, Flash Cache, Flash Pool, FlashRay, FlexArray, FlexCache, FlexClone, FlexPod, FlexScale, FlexShare, FlexVol, FPolicy, GetSuccessful, LockVault, Manage ONTAP, Mars, MetroCluster, MultiStore, NetApp Insight, OnCommand, ONTAP, ONTAPI, RAID DP, SANtricity, SecureShare, Simplicity, Simulate ONTAP, Snap Creator, SnapCopy, SnapDrive, SnapIntegrator, SnapLock, SnapManager, SnapMirror, SnapMover, SnapProtect, SnapRestore, Snapshot, SnapValidator, SnapVault, StorageGRID, Tech OnTap, Unbound Cloud, and WAFL are trademarks or registered trademarks of NetApp, Inc., in the United States, and/or other countries. A current list of NetApp trademarks is available on the web at *http://www.netapp.com/us/legal/netapptmlist.aspx*.

Cisco and the Cisco logo are trademarks of Cisco in the U.S. and other countries. All other brands or products are trademarks or registered trademarks of their respective holders and should be treated as such.

How to send your comments

You can help us to improve the quality of our documentation by sending us your feedback.

Your feedback is important in helping us to provide the most accurate and high-quality information. If you have suggestions for improving this document, send us your comments by email to *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.

You can also contact us in the following ways:

- NetApp, Inc., 495 East Java Drive, Sunnyvale, CA 94089 U.S.
- Telephone: +1 (408) 822-6000
- Fax: +1 (408) 822-4501
- Support telephone: +1 (888) 463-8277

Index