



Full credit is given to all the above companies including the Operating System that this PDF file was generated!

Windows PowerShell Get-Help on Cmdlet 'Connect-VirtualDisk'

PS:>Get-HELP Connect-VirtualDisk -Full

NAME

Connect-VirtualDisk

SYNOPSIS

Connects a disconnected virtual disk to the specified computer when using the Windows Storage subsystem.

SYNTAX

```
Connect-VirtualDisk [-FriendlyName] <String[]> [-AsJob] [-CimSession <CimSession[]>] [-PassThru] [-StorageNodeName <String>] [-ThrottleLimit <Int32>]  
[<CommonParameters>]
```

```
Connect-VirtualDisk [-AsJob] [-CimSession <CimSession[]>] -InputObject <CimInstance[]> [-PassThru]  
[-StorageNodeName <String>] [-ThrottleLimit <Int32>]  
[<CommonParameters>]
```

```
Connect-VirtualDisk [-AsJob] [-CimSession <CimSession[]>] -Name <String[]> [-PassThru] [-StorageNodeName <String>]  
[-ThrottleLimit <Int32>] [<CommonParameters>]
```

```
Connect-VirtualDisk [-AsJob] [-CimSession <CimSession[]>] [-PassThru] [-StorageNodeName <String>] [-ThrottleLimit
```

<Int32> -UniqueId <String[]> [<CommonParameters>]

DESCRIPTION

The Connect-VirtualDisk cmdlet connects a disconnected virtual disk to the specified computer when using the Windows Storage subsystem.

To connect a virtual disk to a remote computer, from a management node in a subsystem that is registered on the management node, specify the StorageNodeName parameter.

To connect to a virtual disk to a remote computer, use the CimSession parameter, or use the Enter-PSSession or Invoke-Command cmdlets in conjunction with

Connect-VirtualDisk .

PARAMETERS

-AsJob [<SwitchParameter>]

Runs the cmdlet as a background job. Use this parameter to run commands that take a long time to complete.

Required? false

Position? named

Default value False

Accept pipeline input? False

Accept wildcard characters? false

-CimSession <CimSession[]>

Runs the cmdlet in a remote session or on a remote computer. Enter a computer name or a session object, such as the output of a New-CimSession

(<https://go.microsoft.com/fwlink/?LinkId=227967>)

or

[Get-CimSession] (<https://go.microsoft.com/fwlink/?LinkId=227966>) cmdlet. The default is the current session on the local computer.

Required? false

Page 2/7

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-FriendlyName <String[]>

Specifies the friendly name of the virtual disk that you want to connect.

Required? true

Position? 0

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-InputObject <CimInstance[]>

Specifies the input object that is used in a pipeline command.

Required? true

Position? named

Default value None

Accept pipeline input? True (ByValue)

Accept wildcard characters? false

-Name <String[]>

Specifies the name of the virtual disk that you want to connect.

Required? true

Position? named

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

Specifies that the cmdlet should output an object representing a virtual disk. By default, this cmdlet does not generate any output.

Required?	false
Position?	named
Default value	False
Accept pipeline input?	False
Accept wildcard characters?	false

-StorageNodeName <String>

Specifies the name of a computer on which to run the cmdlet.

Use this parameter only when you run the cmdlet from a management node to connect a virtual disk to a remote cluster subsystem for the purpose of setting the

virtual disk to manual-attach. To set a virtual disk to manual-attach, use the Set-VirtualDisk cmdlet and specify the IsManualAttach parameter.

Required?	false
Position?	named
Default value	None
Accept pipeline input?	False
Accept wildcard characters?	false

-ThrottleLimit <Int32>

Specifies the maximum number of concurrent operations that can be established to run the cmdlet. If this parameter is omitted or a value of `0` is entered, then

Windows PowerShell calculates an optimum throttle limit for the cmdlet based on the number of CIM cmdlets that are running on the computer. The throttle limit

applies only to the current cmdlet, not to the session or to the computer.

Required?	false
Position?	named
Default value	None

Accept pipeline input? False

Accept wildcard characters? false

-UniqueId <String[]>

Specifies the UniqueID of the virtual disk to connect.

Required? true

Position? named

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

<CommonParameters>

This cmdlet supports the common parameters: Verbose, Debug,

ErrorAction, ErrorVariable, WarningAction, WarningVariable,

OutBuffer, PipelineVariable, and OutVariable. For more information, see

about_CommonParameters (<https://go.microsoft.com/fwlink/?LinkId=113216>).

INPUTS

Microsoft.Management.Infrastructure.CimInstance#ROOT/Microsoft/Windows/Storage/MSFT_VirtualDisk

You can pipe VirtualDisk objects to the InputObject parameter to specify one or more virtual disks to connect to the local computer.

OUTPUTS

Microsoft.Management.Infrastructure.CimInstance#ROOT/Microsoft/Windows/Storage/MSFT_VirtualDisk

The Connect-VirtualDisk cmdlet does not output objects unless you use the Passthru parameter, in which case it outputs objects that represent the virtual disks that you connected.

NOTES

* When used in Failover Cluster, cmdlets from the Storage module operate on cluster level (all servers in the cluster).

--- Example 1: Connect a virtual disk to the local computer ---

```
PS C:\>Connect-VirtualDisk -FriendlyName "VirtualDisk1"
```

This example connects the virtual disk named VirtualDisk1 to the local computer.

Example 2: Connect all disconnected virtual disks on a remote server

```
PS C:\>Get-VirtualDisk -CimSession Srv2 | Where-Object -Filter { $_.OperationalStatus -eq "Detached" } | Connect-VirtualDisk
```

This example uses the Get-VirtualDisk cmdlet to get all virtual disk objects on the computer Srv2 (by using the CimSession parameter). It then pipes the objects to

the Where-Object cmdlet to filter out all virtual disks except those with the "Detached" operational status. Finally, it pipes these objects to the

Connect-VirtualDisk cmdlet.

Example 3: Connect all disconnected virtual disks on two remote computers by using Invoke-Command

```
PS C:\>Invoke-Command -ComputerName Srv1, Srv2 -ScriptBlock { Get-VirtualDisk | Where-Object -Filter { $_.OperationalStatus -eq "Detached" } | Connect-VirtualDisk }
```

This example uses the Invoke-Command cmdlet to run the cmdlets specified in Example 2 (now specified in the ScriptBlock parameter) on multiple remote computers

simultaneously, with all processing done in parallel on each computer and only the results being sent back to the local PowerShell session.

Example 4: Connect a virtual disk to a remote computer from a management node

```
PS C:\>Get-StorageSubSystem -FriendlyName "Clustered Storage Spaces on Cluster01" | Get-VirtualDisk -FriendlyName "VDisk01" | Connect-VirtualDisk -StorageNodeName "ClustNode1"
```

This example connects a virtual disk to a remote computer from a management node in a subsystem that is registered on the management node.

RELATED LINKS

Online

Version:

https://learn.microsoft.com/powershell/module/storage/connect-virtualdisk?view=windowsserver2022-ps&wt.mc_id=ps-gethelp

Invoke-Command <https://go.microsoft.com/fwlink/?LinkID=135225>

Disconnect-VirtualDisk

Get-VirtualDisk

Hide-VirtualDisk

New-VirtualDisk

Remove-VirtualDisk

Repair-VirtualDisk

Resize-VirtualDisk

Set-VirtualDisk

Show-VirtualDisk