



# **Cmdlet Reference for Virtual Machine Manager in System Center 2012 SP1**

---

Microsoft Corporation

Published: January 15, 2013

## **Applies To**

System Center 2012 Service Pack 1 (SP1) - Virtual Machine Manager (VMM)

## **Feedback**

Send suggestions and comments about this guide to [sc2012docs@microsoft.com](mailto:sc2012docs@microsoft.com).

This document is provided "as-is". Information and views expressed in this document, including URL and other Internet Web site references, may change without notice.

Some examples depicted herein are provided for illustration only and are fictitious. No real association or connection is intended or should be inferred.

This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes. You may modify this document for your internal, reference purposes.

This document is confidential and proprietary to Microsoft. It is disclosed and can be used only pursuant to a non-disclosure agreement.

© 2011 Microsoft Corporation. All rights reserved.

Microsoft, Active Directory, Bing, Excel, Hyper-V, Internet Explorer, Silverlight, SQL Server, Windows, Windows Intune, Windows PowerShell, Windows Server, and Windows Vista are trademarks of the Microsoft group of companies. All other trademarks are property of their respective owners.

## Revision History

---

Release Date	Changes
January 15, 2013	Original Release of this guide.

## Contents

Add-SCApplicationDeployment .....	23
Add-SCApplicationHostTemplate .....	31
Add-SCComputerTierTemplate .....	39
Add-SCCustomPlacementRule .....	46
Add-SCLibraryServer.....	52
Add-SCLibraryShare.....	59
Add-SCLoadBalancer .....	65
Add-SCNetworkGateway.....	72
Add-SCNetworkRoute .....	78
Add-SCOperatingSystem .....	81
Add-SCPatch .....	85
Add-SCPowerOptimizationRange .....	89
Add-SCPXE Server .....	94
Add-SCScriptCommand .....	98
Add-SCServerFeature .....	106
Add-SCServicingWindowSubscription.....	110
Add-SCSQLDeployment.....	115
Add-SCSQLScriptCommand .....	124
Add-SCStorageProvider .....	132
Add-SCUpdateServer .....	139
Add-SCVirtualizationManager .....	144
Add-SCVirtualSwitchExtensionManager .....	150
Add-SCVMHost .....	155
Add-SCVMHostCluster .....	175
Add-SCVMHostNetworkAdapter .....	187
Add-SCVMNetworkGateway .....	194

---

Add-SCVPNConnection.....	198
Backup-SCVMMServer.....	208
Clear-SCPowerOptimizationSchedule.....	213
Clear-SCPROTip .....	217
Compress-SCVirtualDiskDrive.....	220
Convert-SCVirtualDiskDrive .....	224
Copy-SCStorageVolume .....	233
Copy-SCVirtualHardDisk .....	243
Disable-SCLoadBalancerVIPMember .....	249
Disable-SCRunAsAccount.....	253
Disable-SCVMHost.....	257
Enable-SCLoadBalancerVIPMember .....	262
Enable-SCRunAsAccount.....	266
Enable-SCVMHost.....	270
Expand-SCVirtualDiskDrive.....	274
Export-SCLibraryPhysicalResource .....	278
Export-SCTemplate .....	282
Find-SCCluster .....	288
Find-SCComputer.....	295
Find-SCJob.....	309
Find-SCLibraryShare .....	313
Get-SCApplicableVMHostGroup .....	316
Get-SCApplication .....	319
Get-SCApplicationDeployment.....	322
Get-SCApplicationHost.....	325
Get-SCApplicationHostTemplate.....	329
Get-SCApplicationPackage .....	333

---

Get-SCApplicationProfile .....	337
Get-SCApplicationSetting .....	341
Get-SCAvailabilitySetName.....	345
Get-SCBaseline .....	347
Get-SCCapabilityProfile .....	350
Get-SCCertificate.....	352
Get-SCCloud .....	356
Get-SCCloudCapacity .....	359
Get-SCCloudProtectionInfo .....	361
Get-SCCloudUsage .....	364
Get-SCClusterVirtualNetwork .....	368
Get-SCComplianceStatus.....	370
Get-SCComputerConfiguration.....	373
Get-SCComputerTier .....	376
Get-SCComputerTierConfiguration .....	379
Get-SCComputerTierTemplate.....	381
Get-SCConfigurationProvider .....	385
Get-SCCPUType .....	388
Get-SCCustomPlacementRule .....	390
Get-SCCustomProperty.....	392
Get-SCCustomPropertyValue.....	395
Get-SCCustomResource .....	398
Get-SCDependentLibraryResource.....	402
Get-SCDirectoryChildItem .....	406
Get-SCDriverPackage .....	409
Get-SCDynamicOptimizationConfiguration .....	413
Get-SCExtensionUplinkPortProfile .....	416

---

Get-SCGatewayConnection .....	419
Get-SCGuestOSProfile .....	421
Get-SCHardwareProfile .....	424
Get-SCHostReserve .....	427
Get-SCIPAddress .....	430
Get-SCISO .....	435
Get-SCISOConfiguration .....	439
Get-SCJob .....	442
Get-SCLibraryRating .....	447
Get-SCLibraryServer .....	450
Get-SCLibraryShare .....	453
Get-SCLoadBalancer .....	456
Get-SCLoadBalancerConfiguration .....	462
Get-SCLoadBalancerTemplate .....	464
Get-SCLoadBalancerVIP .....	468
Get-SCLoadBalancerVIPMember .....	471
Get-SCLoadBalancerVIPTemplate .....	474
Get-SCLogicalNetwork .....	478
Get-SCLogicalNetworkDefinition .....	483
Get-SCLogicalSwitch .....	488
Get-SCMACAddress .....	491
Get-SCMACAddressPool .....	496
Get-SCNativeUplinkPortProfile .....	500
Get-SCNetworkGateway .....	503
Get-SCNotification .....	506
Get-SCOperatingSystem .....	508
Get-SCOpsMgrConnection .....	511

---

Get-SCPendingServiceSetting .....	514
Get-SCPerformanceData.....	516
Get-SCPlacementConfiguration .....	521
Get-SCPortClassification .....	523
Get-SCPowerOptimizationRange .....	525
Get-SCPROMonitor .....	528
Get-SCPROMonitorConfiguration .....	532
Get-SCPROMonitorState.....	538
Get-SCPROTip .....	543
Get-SCPXEserver .....	548
Get-SCRunAsAccount .....	551
Get-SCRunAsAccountConsumer .....	555
Get-SCScript.....	557
Get-SCScriptCommand .....	562
Get-SCScriptCommandSetting.....	565
Get-SCServerFeature .....	567
Get-SCService .....	571
Get-SCServiceConfiguration .....	575
Get-SCServiceSetting.....	578
Get-SCServiceTemplate.....	582
Get-SCServicingWindow .....	585
Get-SCServicingWindowSubscription .....	590
Get-SCSharedResource.....	594
Get-SCSQLDeployment .....	597
Get-SCSQLProfile .....	601
Get-SCSQLScriptCommand.....	605
Get-SCSSASConnection .....	607

---

Get-SCSSHKey .....	609
Get-SCStaticIPAddressPool .....	613
Get-SCStep.....	619
Get-SCStorageArray.....	621
Get-SCStorageClassification .....	626
Get-SCStorageDisk .....	629
Get-SCStorageFileServer.....	632
Get-SCStorageFileShare.....	635
Get-SCStorageLogicalUnit .....	640
Get-SCStoragePool .....	643
Get-SCStorageProvider .....	646
Get-SCStorageVolume .....	649
Get-SCTag.....	653
Get-SCTemplatePackage.....	655
Get-SCUpdate .....	657
Get-SCUpdateServer.....	661
Get-SCUplinkPortProfileSet.....	664
Get-SCUserRole .....	667
Get-SCUserRoleMembership .....	671
Get-SCUserRoleQuota .....	674
Get-SCVirtualCOMPort.....	677
Get-SCVirtualDiskDrive .....	681
Get-SCVirtualDVDDrive.....	685
Get-SCVirtualFloppyDisk.....	689
Get-SCVirtualFloppyDrive .....	693
Get-SCVirtualHardDisk.....	697
Get-SCVirtualHardDiskConfiguration .....	703



---

Get-SCVirtualizationManager .....	706
Get-SCVirtualMachine .....	709
Get-SCVirtualNetwork .....	715
Get-SCVirtualNetworkAdapter .....	719
Get-SCVirtualNetworkAdapterConfiguration .....	725
Get-SCVirtualNetworkAdapterExtensionPortProfile .....	728
Get-SCVirtualNetworkAdapterNativePortProfile.....	731
Get-SCVirtualNetworkAdapterPortProfileSet .....	734
Get-SCVirtualScsiAdapter .....	737
Get-SCVirtualSwitchExtension .....	741
Get-SCVirtualSwitchExtensionManager .....	744
Get-SCVMCheckpoint .....	748
Get-SCVMConfiguration .....	752
Get-SCVMConsoleInformation .....	756
Get-SCVMHost .....	758
Get-SCVMHostCluster.....	763
Get-SCVMHostGroup .....	766
Get-SCVMHostNetworkAdapter .....	769
Get-SCVMHostNetworkAdapterProfile .....	773
Get-SCVMHostProfile.....	775
Get-SCVMHostRating.....	778
Get-SCVMMAccessLicense .....	797
Get-SCVMMDRServiceRegistration .....	800
Get-SCVMMManagedComputer .....	802
Get-SCVMMServer.....	805
Get-SCVMNetwork .....	812
Get-SCVMNetworkGateway .....	815

---

Get-SCVMSubnet .....	817
Get-SCVMTemplate .....	820
Get-SCVMwareResourcePool .....	824
Get-SCVMXComputerConfiguration .....	827
Get-SCVPNConnection .....	829
Grant-SCIPAddress .....	831
Grant-SCMACAddress .....	836
Grant-SCResource .....	842
Import-SCLibraryPhysicalResource .....	848
Import-SCStorageProvider .....	852
Import-SCTemplate .....	855
Install-SCVMHostCluster .....	862
Invoke-SCPROTip .....	870
Invoke-SCScriptCommand .....	873
Mount-SCStorageDisk .....	878
Move-SCVirtualHardDisk .....	887
Move-SCVirtualMachine .....	894
Move-SCVMHost .....	905
Move-SCVMHostCluster .....	909
Move-SCVMHostGroup .....	913
New-SCApplicationProfile .....	917
New-SCBaseline .....	923
New-SCCapabilityProfile .....	927
New-SCCloud .....	949
New-SCComputerConfiguration .....	955
New-SCCustomProperty .....	960
New-SCDefaultGateway .....	965

---

New-SCExternalJob .....	968
New-SCGuestOSProfile .....	971
New-SCHardwareProfile.....	986
New-SCLoadBalancerConnectionPersistence .....	1006
New-SCLoadBalancerHealthMonitor .....	1008
New-SCLoadBalancerProtocol .....	1012
New-SCLoadBalancerTemplate .....	1015
New-SCLoadBalancerVIP .....	1021
New-SCLoadBalancerVIPMember .....	1028
New-SCLoadBalancerVIPTemplate .....	1033
New-SCLoadBalancingMethod .....	1040
New-SCLogicalNetwork.....	1042
New-SCLogicalNetworkDefinition.....	1047
New-SCLogicalSwitch .....	1052
New-SCMACAddressPool.....	1057
New-SCNativeUplinkPortProfile .....	1062
New-SCOpsMgrConnection .....	1067
New-SCP2V.....	1072
New-SCPpackageMapping.....	1098
New-SCPortClassification.....	1101
New-SCRunAsAccount.....	1104
New-SCScriptCommandSetting .....	1109
New-SCService .....	1115
New-SCServiceConfiguration.....	1120
New-SCServiceTemplate .....	1127
New-SCServicingWindow.....	1133
New-SCSQLProfile.....	1145

---

New-SCSSASConnection.....	1150
New-SCStaticIPAddressPool.....	1155
New-SCStorageClassification.....	1165
New-SCStorageFileShare .....	1169
New-SCStorageLogicalUnit.....	1175
New-SCSubnetVlan.....	1181
New-SCUplinkPortProfileSet .....	1184
New-SCUserRole .....	1189
New-SCV2V.....	1195
New-SCVirtualDiskDrive.....	1216
New-SCVirtualDVDDrive .....	1236
New-SCVirtualMachine.....	1245
New-SCVirtualNetwork.....	1284
New-SCVirtualNetworkAdapter .....	1294
New-SCVirtualNetworkAdapterNativePortProfile .....	1312
New-SCVirtualNetworkAdapterPortProfileSet .....	1320
New-SCVirtualScsiAdapter .....	1326
New-SCVMCheckpoint .....	1333
New-SCVMConfiguration.....	1338
New-SCVMHost.....	1345
New-SCVMHostConfig .....	1357
New-SCVMHostGroup.....	1366
New-SCVMHostNetworkAdapterConfig .....	1371
New-SCVMHostNetworkAdapterProfile .....	1382
New-SCVMHostProfile .....	1391
New-SCVMNetwork.....	1402
New-SCVMSubnet.....	1410

---

New-SCVMTemplate .....	1416
New-SCVMXComputerConfiguration .....	1449
Publish-SCWindowsPE .....	1453
Read-SCGuestInfo .....	1458
Read-SCLibraryShare .....	1462
Read-SCLoadBalancer .....	1466
Read-SCLoadBalancerVIP .....	1469
Read-SCNetworkGateway.....	1472
Read-SCService .....	1475
Read-SCServiceTemplate .....	1478
Read-SCStorageProvider .....	1481
Read-SCVirtualizationManager .....	1484
Read-SCVirtualMachine .....	1487
Read-SCVirtualSwitchExtensionManager .....	1491
Read-SCVMHost .....	1494
Read-SCVMHostCluster.....	1498
Read-SCVMNetworkGateway .....	1501
Register-SCStorageFileShare .....	1504
Register-SCStorageLogicalUnit.....	1509
Register-SCVirtualMachine .....	1514
Register-SCVMHost .....	1519
Register-SCVMMAccessLicense.....	1525
Register-SCVMMManagedComputer .....	1528
Remove-SCApplicationDeployment .....	1532
Remove-SCApplicationHostTemplate .....	1535
Remove-SCApplicationPackage .....	1539
Remove-SCApplicationProfile .....	1543

---

Remove-SCBaseline .....	1547
Remove-SCCapabilityProfile .....	1551
Remove-SCCloud.....	1555
Remove-SCComputerConfiguration .....	1559
Remove-SCComputerTierTemplate .....	1563
Remove-SCCustomPlacementRule .....	1567
Remove-SCCustomProperty .....	1571
Remove-SCCustomPropertyValue .....	1575
Remove-SCCustomResource .....	1579
Remove-SCDriverPackage.....	1583
Remove-SCGuestOSProfile .....	1587
Remove-SCHardwareProfile .....	1591
Remove-SCISO .....	1594
Remove-SCLibraryServer.....	1599
Remove-SCLibraryShare.....	1604
Remove-SCLoadBalancer .....	1608
Remove-SCLoadBalancerTemplate .....	1612
Remove-SCLoadBalancerVIP .....	1616
Remove-SCLoadBalancerVIPMember .....	1620
Remove-SCLoadBalancerVIPTemplate .....	1624
Remove-SCLogicalNetwork.....	1628
Remove-SCLogicalNetworkDefinition .....	1632
Remove-SCLogicalSwitch .....	1636
Remove-SCMACAddressPool.....	1640
Remove-SCNativeUplinkPortProfile .....	1644
Remove-SCNetworkGateway.....	1648
Remove-SCOperatingSystem .....	1652

---

Remove-SCOpsMgrConnection .....	1656
Remove-SCPortClassification .....	1660
Remove-SCPXEserver .....	1664
Remove-SCRunAsAccount .....	1668
Remove-SCScript .....	1672
Remove-SCScriptCommand .....	1677
Remove-SCServerFeature .....	1681
Remove-SCService .....	1685
Remove-SCServiceConfiguration .....	1689
Remove-SCServiceTemplate .....	1693
Remove-SCServicingWindow .....	1697
Remove-SCServicingWindowSubscription .....	1701
Remove-SCSQLDeployment .....	1706
Remove-SCSQLProfile .....	1709
Remove-SCSQLScriptCommand .....	1712
Remove-SCSSASConnection .....	1716
Remove-SCSSHKey .....	1719
Remove-SCStaticIPAddressPool .....	1723
Remove-SCStorageClassification .....	1727
Remove-SCStorageFileShare .....	1730
Remove-SCStorageLogicalUnit .....	1734
Remove-SCStorageProvider .....	1738
Remove-SCUpdateServer .....	1741
Remove-SCUplinkPortProfileSet .....	1745
Remove-SCUserRole .....	1749
Remove-SCVirtualDiskDrive .....	1752
Remove-SCVirtualDVDDrive .....	1758

---

Remove-SCVirtualFloppyDisk .....	1763
Remove-SCVirtualHardDisk .....	1767
Remove-SCVirtualizationManager .....	1772
Remove-SCVirtualMachine .....	1776
Remove-SCVirtualNetwork .....	1781
Remove-SCVirtualNetworkAdapter .....	1785
Remove-SCVirtualNetworkAdapterNativePortProfile .....	1791
Remove-SCVirtualNetworkAdapterPortProfileSet .....	1795
Remove-SCVirtualScsiAdapter .....	1799
Remove-SCVirtualSwitchExtensionManager .....	1803
Remove-SCVMCheckpoint .....	1807
Remove-SCVMConfiguration .....	1811
Remove-SCVMHost .....	1815
Remove-SCVMHostCluster .....	1821
Remove-SCVMHostGroup .....	1826
Remove-SCVMHostNetworkAdapter .....	1830
Remove-SCVMHostNetworkAdapterProfile .....	1834
Remove-SCVMHostProfile .....	1837
Remove-SCVMNetwork .....	1841
Remove-SCVMNetworkGateway .....	1845
Remove-SCVMSubnet .....	1849
Remove-SCVMTemplate .....	1853
Remove-SCVMXComputerConfiguration .....	1858
Remove-SCVPNConnection .....	1862
Repair-SCStorageFileShare .....	1866
Repair-SCVirtualMachine .....	1870
Repair-SCVirtualNetwork .....	1876



---

Repair-SCVirtualNetworkAdapter .....	1880
Repair-SCVMHost .....	1884
Reset-SCPROMonitorState .....	1888
Reset-SCVirtualMachine .....	1890
Resolve-SCServiceTemplate .....	1893
Restart-SCJob .....	1897
Restart-SCVMHost .....	1900
Restore-SCVMCheckpoint .....	1904
Resume-SCService .....	1908
Resume-SCVirtualMachine .....	1911
Revoke-SCIPAddress .....	1915
Revoke-SCMACAddress .....	1919
Revoke-SCResource .....	1923
Save-SCVirtualMachine .....	1928
Set-SCApplicationDeployment .....	1934
Set-SCApplicationHostTemplate .....	1941
Set-SCApplicationPackage .....	1949
Set-SCApplicationProfile .....	1956
Set-SCApplicationSetting .....	1962
Set-SCBaseline .....	1967
Set-SCCapabilityProfile .....	1974
Set-SCCloud .....	1995
Set-SCCloudCapacity .....	2003
Set-SCComplianceStatus .....	2011
Set-SCComputerTier .....	2016
Set-SCComputerTierConfiguration .....	2021
Set-SCComputerTierTemplate .....	2025

---

Set-SCCustomPlacementRule .....	2032
Set-SCCustomProperty .....	2038
Set-SCCustomPropertyValue .....	2044
Set-SCCustomResource .....	2048
Set-SCDriverPackage.....	2055
Set-SCDynamicOptimizationConfiguration.....	2059
Set-SCExternalJob .....	2066
Set-SCGuestInfo.....	2070
Set-SCGuestOSProfile .....	2076
Set-SCHardwareProfile .....	2090
Set-SCHostReserve .....	2109
Set-SCInternetSCSIHba .....	2121
Set-SCIPAddress.....	2125
Set-SCISO .....	2129
Set-SCISOConfiguration.....	2136
Set-SCLibraryServer.....	2140
Set-SCLibraryShare.....	2147
Set-SCLoadBalancer .....	2151
Set-SCLoadBalancerConfiguration .....	2159
Set-SCLoadBalancerTemplate .....	2165
Set-SCLoadBalancerVIPTemplate .....	2169
Set-SCLogicalNetwork.....	2176
Set-SCLogicalNetworkDefinition .....	2181
Set-SCLogicalSwitch .....	2186
Set-SCMACAddressPool.....	2192
Set-SCNativeUplinkPortProfile .....	2197
Set-SCNetworkGateway.....	2203

---

Set-SCNotification .....	2209
Set-SCOpsMgrConnection .....	2213
Set-SCPackageMapping .....	2217
Set-SCPlacementConfiguration.....	2220
Set-SCPortClassification .....	2229
Set-SCPROMonitorConfiguration .....	2233
Set-SCPROTip .....	2238
Set-SCRunAsAccount .....	2244
Set-SCScript .....	2250
Set-SCScriptCommand .....	2257
Set-SCScriptCommandSetting .....	2264
Set-SCService .....	2271
Set-SCServiceConfiguration.....	2279
Set-SCServiceSetting .....	2285
Set-SCServiceTemplate .....	2290
Set-SCServicingWindow.....	2296
Set-SCSQLDeployment.....	2307
Set-SCSQLProfile.....	2316
Set-SCSQLScriptCommand .....	2321
Set-SCSSHKey.....	2329
Set-SCStaticIPAddressPool .....	2336
Set-SCStorageArray .....	2344
Set-SCStorageClassification .....	2350
Set-SCStorageFileServer .....	2354
Set-SCStorageFileShare .....	2358
Set-SCStorageLogicalUnit.....	2362
Set-SCStoragePool .....	2367

---

Set-SCStorageProvider .....	2372
Set-SCStorageVolume .....	2377
Set-SCUpdate.....	2381
Set-SCUpdateServer .....	2385
Set-SCUplinkPortProfileSet.....	2393
Set-SCUserRole .....	2399
Set-SCUserRoleQuota .....	2412
Set-SCVirtualCOMPort .....	2421
Set-SCVirtualDiskDrive.....	2429
Set-SCVirtualDVDDrive .....	2436
Set-SCVirtualFloppyDisk .....	2446
Set-SCVirtualFloppyDrive.....	2454
Set-SCVirtualHardDisk .....	2459
Set-SCVirtualHardDiskConfiguration.....	2468
Set-SCVirtualizationManager .....	2476
Set-SCVirtualMachine .....	2481
Set-SCVirtualNetwork.....	2516
Set-SCVirtualNetworkAdapter .....	2522
Set-SCVirtualNetworkAdapterConfiguration.....	2538
Set-SCVirtualNetworkAdapterNativePortProfile .....	2547
Set-SCVirtualNetworkAdapterPortProfileSet.....	2555
Set-SCVirtualScsiAdapter.....	2562
Set-SCVirtualSwitchExtension.....	2568
Set-SCVirtualSwitchExtensionManager .....	2572
Set-SCVMCheckpoint.....	2578
Set-SCVMConfiguration .....	2582
Set-SCVMHost .....	2592

---

Set-SCVMHostCluster .....	2614
Set-SCVMHostGroup .....	2620
Set-SCVMHostNetworkAdapter .....	2625
Set-SCVMHostProfile .....	2635
Set-SCVMMServer .....	2646
Set-SCVMNetwork.....	2655
Set-SCVMNetworkGateway .....	2660
Set-SCVMSubnet .....	2664
Set-SCVMTemplate .....	2669
Set-SCVPNConnection.....	2703
Start-SCComplianceScan.....	2713
Start-SCDynamicOptimization .....	2717
Start-SCService .....	2720
Start-SCUpdateRemediation .....	2723
Start-SCUpdateServerSynchronization .....	2731
Start-SCVirtualMachine .....	2735
Start-SCVMHost .....	2739
Stop-SCJob.....	2742
Stop-SCService .....	2744
Stop-SCVirtualMachine .....	2749
Stop-SCVMHost .....	2755
Suspend-SCService .....	2759
Suspend-SCVirtualMachine.....	2762
Test-SCCapabilityProfile.....	2765
Test-SCDomainCredential.....	2768
Test-SCLoadBalancer .....	2771
Test-SCNetworkGateway .....	2776

---

Test-SCPROTip.....	2780
Test-SCServiceTemplate.....	2782
Test-SCVirtualDiskDrive.....	2785
Test-SCVirtualSwitchExtensionManager.....	2789
Test-SCVMHostCluster.....	2793
Uninstall-SCVMHostCluster.....	2798
Unregister-SCStorageFileShare.....	2802
Unregister-SCStorageLogicalUnit.....	2807
Update-SCService.....	2812
Update-SCServiceConfiguration.....	2817
Update-SCVMConfiguration.....	2820
Update-SCVMMManagedComputer.....	2824
Write-SCOpsMgrConnection.....	2828

# Add-SCApplicationDeployment

---

## Add-SCApplicationDeployment

Adds an application to an application profile.

### Syntax

Parameter Set: ServerAppV

```
Add-SCApplicationDeployment -ApplicationPackage <ApplicationPackage> -ApplicationProfile <ApplicationProfile> -Name <String> -ServerAppV [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: SQLDac

```
Add-SCApplicationDeployment -ApplicationPackage <ApplicationPackage> -ApplicationProfile <ApplicationProfile> -Name <String> -SQLDac -SQLDeploymentRunAsAccount <VMMSCredential> [-BlockOnChanges <Boolean> ] [-DACInstanceName <String> ] [-IgnoreDataLoss <Boolean> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RollbackOnFailure <Boolean> ] [-RunAsynchronously] [-SkipPolicyValidation <Boolean> ] [-SQLAuthenticationType <String> ] [-SQLInstanceName <String> ] [-UninstallMode <String> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: WebDeploy

```
Add-SCApplicationDeployment -ApplicationPackage <ApplicationPackage> -ApplicationProfile <ApplicationProfile> -Name <String> -WebDeploy [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

## Detailed Description

The Add-SCApplicationDeployment cmdlet adds an application to an application profile.

## Parameters

### **-ApplicationPackage<ApplicationPackage>**

Specifies an application package object.

Aliases	none
Required?	true
Position?	named
Default Value	none

Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-ApplicationProfile<ApplicationProfile>**

Specifies an application profile object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-BlockOnChanges<Boolean>**

Indicates that the SQL DAC update is blocked if the database schema is different than that defined in the previous DAC.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-DACInstanceName<String>**

Specifies the name of a data-tier application (DAC) instance.

Aliases	none
Required?	false
Position?	named



---

Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-IgnoreDataLoss<Boolean>**

Indicates that data loss which may occur when updating the SQL Server database is ignored.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	true
Position?	named

Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-RollbackOnFailure<Boolean>**

Rolls back any changes made if the SQL Server database update fails.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false
Position?	named

---

Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-ServerAppV**

Indicates that the application is a virtual application.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-SkipPolicyValidation<Boolean>**

Indicates whether policy validation against the SQL Server database should occur.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-SQLAuthenticationType<String>**

Specifies the SQL Server authentication type. Valid values are: SQLServerAuthentication, WindowsAuthentication.

Aliases	none
Required?	false

---

Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-SQLDac**

Indicates that the application is a SQL Server data-tier application (DAC).

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-SQLDeploymentRunAsAccount<VMMSCredential>**

Specifies a Run As account to use to communicate with a SQL Server deployment.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-SQLInstanceName<String>**

Specifies the name of a SQL Server instance.

Aliases	none
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-UninstallMode<String>**

Specifies the uninstall mode. Valid values are: MakeUnmanaged, DetachDatabase, DropDatabase.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-WebDeploy**

Indicates that the application is a Web application.

Aliases	none
Required?	true

---

Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **ApplicationDeployment**

## Examples

### 1: Add a Web application to an application deployment.

The first command gets the application profile object named SvcWebAppProfile01 and stores the object in the \$AppProfile variable.

The second command gets the application package object named WebApp01.zip from the VMM library and stores the object in the \$AppPackage variable.

The last command adds the application package stored in \$AppPackage to the application profile stored in \$AppProfile, and names the application deployment SvcWebDepAD.

```
PS C:\> $AppProfile = Get-SCApplicationProfile -Name "SvcWebAppProfile01"
```

```
PS C:\> $AppPackage = Get-SCApplicationPackage -Name "WebApp01.zip"
```

```
PS C:\> $AppDeployment = Add-SCApplicationDeployment -ApplicationProfile $AppProfile -  
WebDeploy -Name "SvcWebDeployment01" -ApplicationPackage $AppPackage
```

## Related topics

[Get-SCApplicationDeployment](#)

[Remove-SCApplicationDeployment](#)

[Set-SCApplicationDeployment](#)

---

# Add-SCApplicationHostTemplate

---

## Add-SCApplicationHostTemplate

Adds an application host template to a service template.

### Syntax

Parameter Set: DefaultParamSet

```
Add-SCApplicationHostTemplate [-Name] <String> -ApplicationProfile <ApplicationProfile> -  
ComputerName <String> -ServiceTemplate <ServiceTemplate> [-DeploymentOrder <Int32> ] [-  
Description <String> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-  
ServicingOrder <Int32> ] [-Tag <String> ] [ <CommonParameters>]
```

Parameter Set: WebApplicationHost

```
Add-SCApplicationHostTemplate [-Name] <String> -ApplicationProfile <ApplicationProfile> -  
ComputerName <String> -RunAsAccount <VMMSCredential> -ServiceTemplate <ServiceTemplate> -Site  
<String> -WebApplicationHost [-AllowUntrustedServerCertificate <Boolean> ] [-  
AuthenticationType <String> ] [-DeploymentOrder <Int32> ] [-Description <String> ] [-  
JobVariable <String> ] [-Port <Int32> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-  
ServicingOrder <Int32> ] [-Tag <String> ] [ <CommonParameters>]
```

### Detailed Description

The Add-SCApplicationHostTemplate adds an application host template to a service template. An application host template is used to deploy a SQL data-tier application (DAC) on a deployed SQL server or a Web Deploy package to a supported Web Application Host.

### Parameters

#### **-AllowUntrustedServerCertificate<Boolean>**

Specifies that the deployment may proceed when the target deployment server presents an untrusted server certificate. This parameter is used in conjunction with a Web Application host template.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-ApplicationProfile<ApplicationProfile>**

Specifies an application profile object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-AuthenticationType<String>**

Specifies the authentication mechanism used to facilitate the deployment of web applications associated with the template. This parameter is used in conjunction with a Web Application host template. Valid values are: NTLM and Basic.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-ComputerName<String>**

Specifies the name of a computer that VMM can uniquely identify on your network. Valid formats are: FQDN, IPv4 or IPv6 address, or NetBIOS name.

NOTE: See the examples for a specific cmdlet to determine how that cmdlet specifies the computer name.



---

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-DeploymentOrder<Int32>**

Specifies the order in which a computer tier, application host, or application is deployed.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Description<String>**

States a description for the specified object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Port<Int32>**

Specifies the network port to use when adding an object or creating a connection. Valid values are: 1 to 4095.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-RunAsAccount<VMMSCredential>**

Specifies a Run As account that contains credentials with permission to perform this action.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-ServiceTemplate<ServiceTemplate>**

Specifies a service template object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-ServicingOrder<Int32>**

Specifies the order in which a computer tier or application host is serviced.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Site<String>**

Specifies the web site to which any associated web deploy packages are deployed. This parameter is used in conjunction with a Web Application host template,

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-Tag<String>**

Specifies a word or phrase to associate with an object so that you can search for all objects with the specified set of tags. You can search for a subset of tags, or you can search for the full set of tags.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-WebApplicationHost**

Specifies that the application host template is used for a Web Application Host.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **ApplicationHostTemplate**

---

## Examples

### 1: Add a SQL application host template to a service template.

The first command gets the application profile object named SvcWebAppProfile01 and stores the object in the \$AppProfile variable.

The second command gets the service template object named ServiceTemplate01 and stores the object in the \$ServiceTemplate variable.

The third command adds an application host template to the service template stored in \$ServiceTemplate

```
PS C:\> $AppProfile = Get-SCApplicationProfile -Name "SvcWebAppProfile01"
PS C:\> $ServiceTemplate = Get-SCServiceTemplate -Name "ServiceTemplate01"
PS C:\> Add-SCApplicationHostTemplate -Name "SQL Application Host" -ComputerName
"SQLServer01.Contoso.com" -ApplicationProfile $AppProfile -ServiceTemplate $ServiceTemplate
```

### 2: Add a web application host template to a service template.

The first command gets the application profile object named WebAppProfile01 and stores the object in the \$AppProfile variable.

The second command gets the service template object named ServiceTemplate01 and stores the object in the \$ServiceTemplate variable.

The third command gets the Run As Account object named IIS Site Admin and stores the object in the \$WebRAA variable.

The last command adds a web application host template to the service template stored in \$ServiceTemplate (in this case, ServiceTemplate01).

```
PS C:\> $AppProfile = Get-SCApplicationProfile -Name "WebAppProfile01"
PS C:\> $ServiceTemplate = Get-SCServiceTemplate -Name "ServiceTemplate01"
PS C:\> $WebRAA = Get-SCRunAsAccount -Name "IIS Site Admin"
PS C:\> $WebAppHostTemplate = Add-SCApplicationHostTemplate -Name "Web Farm" -
ServiceTemplate $ServiceTemplate -ApplicationProfile -$AppProfile -ComputerName "@Web
Server@" -DeploymentOrder 2 -WebAppHost -Port 8172 -Site "Default Web Site" -
AuthenticationType "NTLM" -AllowUntrustedServerCertificate -RunAsAccount $WebRAA
```

## Related topics

[Get-SCApplicationHostTemplate](#)

[Get-SCApplicationProfile](#)

[Get-SCServiceTemplate](#)

[Remove-SCApplicationHostTemplate](#)

[Set-SCApplicationHostTemplate](#)

---

# Add-SCComputerTierTemplate

---

## Add-SCComputerTierTemplate

Adds a computer tier template to a service template.

### Syntax

Parameter Set: Default

```
Add-SCComputerTierTemplate [-Name] <String> -ServiceTemplate <ServiceTemplate> -VMTemplate <Template> [-BlockAutomaticMigration <Boolean> ] [-DefaultInstanceCount <Int32> ] [-DeploymentOrder <Int32> ] [-Description <String> ] [-EnableAvailabilitySet <Boolean> ] [-InstanceMaximumCount <Int32> ] [-InstanceMinimumCount <Int32> ] [-JobVariable <String> ] [-NumberOfUpgradeDomains <Int32> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-ServicingOrder <Int32> ] [-Tag <String> ] [ <CommonParameters>]
```

### Detailed Description

The Add-SCComputerTierTemplate cmdlet adds a computer tier template to a service template. A computer tier template contains a virtual machine template that is used to create a virtual machine.

For information about service templates, type: "Get-Help New-SCServiceTemplate -detailed". For more information about virtual machine templates, type: "Get-Help New-SCVMTemplate -detailed".

### Parameters

#### **-BlockAutomaticMigration<Boolean>**

Indicates whether the computer can be automatically migrated. When set to \$True, automatic migration is blocked. When set to \$False, automatic migration is allowed. Default value: \$False.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-DefaultInstanceCount<Int32>**

Specifies the default instance count for a computer tier that can be scaled out. Default value: 1.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-DeploymentOrder<Int32>**

Specifies the order in which a computer tier, application host, or application is deployed.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Description<String>**

States a description for the specified object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false



---

### **-EnableAvailabilitySet<Boolean>**

Indicates whether an availability set is generated on a service tier when a service is deployed.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-InstanceMaximumCount<Int32>**

Specifies the maximum number of virtual machines to which a service instance can scale out.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-InstanceMinimumCount<Int32>**

Specifies the minimum number of virtual machines to which a service instance can scale in.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-NumberOfUpgradeDomains<Int32>**

Specifies the number of upgrade domains for a computer tier that can be scaled out. Default value: 1.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-ServiceTemplate<ServiceTemplate>**

Specifies a service template object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

---

## **-ServicingOrder<Int32>**

Specifies the order in which a computer tier or application host is serviced.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Tag<String>**

Specifies a word or phrase to associate with an object so that you can search for all objects with the specified set of tags. You can search for a subset of tags, or you can search for the full set of tags.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMTemplate<Template>**

Specifies a VMM template object used to create virtual machines.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **ComputerTierTemplate**

## Examples

### 1: Add a computer tier template to a service template.

The first command gets the virtual machine template object named WebTemplate01 and stores the object in the \$WebTemplate variable.

The second command gets the service template object named ServiceTemplate01 and stores the object in the \$ServiceTemplate variable.

The last command adds a computer tier template to the service template stored in \$ServiceTemplate.

```
PS C:\> $WebTemplate = Get-SCVMTemplate | where { $_.Name -eq "WebTemplate01" }
PS C:\> $ServiceTemplate = Get-SCServiceTemplate -Name "ServiceTemplate01"
PS C:\> Add-SCComputerTierTemplate -ServiceTemplate $ServiceTemplate -VMTemplate
$WebTemplate -Name "Web Tier" -DefaultInstanceCount 3 -InstanceMinimumCount 1 -
InstanceMaximumCount 5 -DeploymentOrder 1 -ServicingOrder 1 -NumberOfUpgradeDomains 1
```

## Related topics

[Get-SCComputerTierTemplate](#)

[Get-SCServiceTemplate](#)

[Get-SCVMTemplate](#)

[Remove-SCComputerTierTemplate](#)

[Set-SCComputerTierTemplate](#)

---

# Add-SCCustomPlacementRule

---

## Add-SCCustomPlacementRule

Adds a custom placement rule to the placement configuration for a host group.

### Syntax

Parameter Set: MustMatch

```
Add-SCCustomPlacementRule -CustomPropertyName <String> -MustMatch -PlacementConfiguration <PlacementConfigurationSettings> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [-Confirm] [-WhatIf] [<CommonParameters>]
```

Parameter Set: MustNotMatch

```
Add-SCCustomPlacementRule -CustomPropertyName <String> -MustNotMatch -PlacementConfiguration <PlacementConfigurationSettings> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [-Confirm] [-WhatIf] [<CommonParameters>]
```

Parameter Set: ShouldMatch

```
Add-SCCustomPlacementRule -CustomPropertyName <String> -PlacementConfiguration <PlacementConfigurationSettings> -ShouldMatch [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [-Confirm] [-WhatIf] [<CommonParameters>]
```

Parameter Set: ShouldNotMatch

```
Add-SCCustomPlacementRule -CustomPropertyName <String> -PlacementConfiguration <PlacementConfigurationSettings> -ShouldNotMatch [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [-Confirm] [-WhatIf] [<CommonParameters>]
```

### Detailed Description

The Add-SCCustomPlacementRule adds a custom placement rule to the placement configuration for a host group.

### Parameters

#### **-CustomPropertyName<String>**

Specifies the name for a custom property.

---

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-MustMatch**

Indicates that the property value of the virtual machine must match the host.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-MustNotMatch**

Indicates that the property value of the virtual machine must not match the host.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-PlacementConfiguration<PlacementConfigurationSettings>**

Specifies a placement configuration object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.



---

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-ShouldMatch**

Indicates that the property value of the virtual machine should match the host.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-ShouldNotMatch**

Indicates that the property value of the virtual machine should not match the host.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

---

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Confirm**

Prompts you for confirmation before executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

---

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **CustomPlacementRule**

## Examples

### 1: Add a new custom placement rule to a placement configuration for a host group

The first command gets the host group object named HostGroup01 and stores the object in the \$HostGroup variable.

The second command gets the placement configuration object for the host group stored in \$HostGroup and places the object in the \$PlacementConfig variable.

The third command gets the custom property object named Cost Center and stores the object in the \$CustomProp variable.

The last command adds a custom placement rule to the placement configuration stored in \$PlacementConfig.

```
PS C:\> $HostGroup = Get-SCVMHostGroup "HostGroup01"
```

```
PS C:\> $PlacementConfig = Get-SCPlacementConfiguration -VMHostGroup $HostGroup
```

```
PS C:\> $CustomProp = Get-SCCustomProperty -Name "Cost Center"
```

```
PS C:\> Add-SCCustomPlacementRule -PlacementConfiguration $PlacementConfig -MustMatch -  
CustomProperty $CustomProp
```

---

# Add-SCLibraryServer

---

## Add-SCLibraryServer

Adds a computer as a library server to VMM.

### Syntax

Parameter Set: Default

```
Add-SCLibraryServer [-ComputerName] <String> -Credential <VMMCredential> [-Description <String> ] [-EnableUnencryptedFileTransfer <Boolean> ] [-JobGroup <Guid> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMHostGroup <HostGroup> ] [-VMMServer <ServerConnection> ] [ <CommonParameters> ]
```

### Detailed Description

The Add-SCLibraryServer cmdlet adds one or more computers as library servers to Virtual Machine Manager (VMM). For a computer to be a library server, it must be in the same domain as, or in a trusted domain with, the VMM server. For VMM library server system requirements, see "System Requirements: VMM Library Server" in the TechNet Library at <http://go.microsoft.com/fwlink/?LinkID=213754>.

When you add a computer as a library server to VMM, VMM automatically installs the Virtual Machine Manager Agent software on that server.

The VMM library is made up of two primary components:

- LIBRARY. The portion of the VMM database that stores objects that represent all library resources.
- LIBRARY RESOURCE FILES. Files that are stored in library shares on one or more physical library servers. Library resources can be distributed across multiple physical library servers. Some library objects have files and others do not.

VMM library resources include virtual machine templates, hardware profiles, guest operating system profiles, virtual hard disks (Windows-based .vhd files, Citrix XenServer-based .vhd files or VMware-based .vmdk files), virtual floppy disks (Windows-based .vfd files or VMware-based .flp files), ISO images (.iso files), and scripts. In addition, you can store virtual machines in the library that you do not want deployed on a host.

Of these resources, templates, hardware profiles, and guest operating system profiles are represented only by objects stored in the library. The other resources are files stored in the file system on library servers and objects that correspond to those files stored in the library.

---

## Parameters

### **-ComputerName<String>**

Specifies the name of a computer that VMM can uniquely identify on your network. Valid formats are: FQDN, IPv4 or IPv6 address, or NetBIOS name.

NOTE: See the examples for a specific cmdlet to determine how that cmdlet specifies the computer name.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-Credential<VMMSecurityCredential>**

Specifies a credential object or, for some cmdlets, a Run As account object that contains the user name and password of an account that has permission to perform this action. Or, in the case of Restart-SCJob, has permission to complete a restarted task.

For more information about the PSCredential object, type: "Get-Help Get-Credential".

For more information about Run As accounts, type: "Get-Help New-SCRunAsAccount".

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Description<String>**

States a description for the specified object.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-EnableUnencryptedFileTransfer<Boolean>**

Indicates, when set to True, that network file transfers do not require encryption. Allowing unencrypted network file transfers can improve performance if neither the source host nor the destination host requires encryption.

Use this parameter to:

- Enable unencrypted file transfers into, or out of, the library.
- Enable unencrypted file transfers into, out of, or within a host group.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-JobGroup<Guid>**

Specifies an identifier for a series of commands that will run as a set just before the final command that includes the same job group identifier runs.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-VMHostGroup<HostGroup>**

Specifies a virtual machine host group object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **LibraryServer**



---

## Examples

### 1: Add a library server.

The first command prompts you for credentials. When the dialog box appears, type the user name and password for either a local Administrator account or a domain account with administrator rights on the library server.

The second command adds the library server object named LibraryServer01 to the library on VMMServer01.

```
PS C:\> $Creds = Get-Credential
PS C:\> Add-SCLibraryServer -VMMServer "VMMServer01.Contoso.com" -ComputerName
"LibraryServer01.Contoso.com" -Credential $Creds -RunAsynchronously
```

### 2: Add a highly available file server with two nodes as a library server.

This example assumes the following: you have created a cluster with at least two nodes, you have created a highly available file server, and you have created a share on the highly available file server (in this example, this is represented by \\HAFfileServer01.Contoso.com\LibShare).

The first command uses Get-Credential to prompt you to supply a user name and password and stores your credentials in \$Credential. The required credentials for this operation are a domain account with administrator rights on each node of a failover cluster hosting the highly available file server that you want to add to VMM.

The second command uses the Find-SCCluster cmdlet to confirm that HAFfileServer01 is a highly available file server and stores the cluster object in the \$Cluster variable.

The third command uses a foreach loop to pass each cluster node to Add-SCLibraryServer, which adds the nodes as library servers. For more information about the Windows PowerShell foreach loop statement, type: "Get-Help about\_ForEach".

The fourth command uses Add-SCLibraryServer to add the highly available file server named HAFfileServer01 to VMM as a library server.

The last command uses Add-SCLibraryShare to add the specified share on the highly available file server. For more information about adding library shares, type: "Get-Help Add-SCLibraryShare".

```
PS C:\> $Credential = Get-Credential
PS C:\> $Cluster = Find-SCCluster -ComputerName "HAFfileServer01.Contoso.com" -Credential
$Credential
PS C:\> ForEach ($Node in $Cluster.ClusterNodes) { Add-SCLibraryServer -ComputerName $Node -
Credential $Credential}
PS C:\> Add-SCLibraryServer -ComputerName "HAFfileServer01.Contoso.com" -Credential
$Credential
PS C:\> Add-SCLibraryShare -SharePath "\\HAFfileServer01.Contoso.com\LibShare" -Credential
$Credential
```

---

## Related topics

[Add-SCLibraryShare](#)

[Get-SCLibraryServer](#)

[Get-SCVMMServer](#)

[Remove-SCLibraryServer](#)

[Set-SCLibraryServer](#)

---

# Add-SCLibraryShare

---

## Add-SCLibraryShare

Adds Windows shares on the file system of a library server to the VMM library as library shares.

### Syntax

Parameter Set: Default

```
Add-SCLibraryShare [-SharePath] <String> [-AddDefaultResources] [-Credential <PSCredential> ] [-Description <String> ] [-JobGroup <Guid> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Add-SCLibraryShare cmdlet adds Windows shares on the file system of a library server to the VMM library as library shares.

Before you can add a library share to the VMM library, you must first create the share in the Windows file system. You can, for example, use Windows Explorer to create and share a folder that you want to add to the library.

If you create a Windows share at the same level as the default library share (MSSCVMMLibrary) created by VMM Setup or on a separate library server, use the Add-SCLibraryShare cmdlet to add that share to the VMM library.

If you create a Windows folder under the default VMM library share (MSSCVMMLibrary), VMM automatically scans the share, discovers all existing objects stored on that share that qualify as library objects, and adds the library objects to the library. However, you can use the Read-SCLibraryShare cmdlet to manually refresh that share and import its contents into the VMM library.

NOTE: Library resources that can be discovered only by the library refresher but not created by an administrator include virtual hard disks (Windows-based .vhd files, Citrix XenServer-based .vhd files or VMware-based .vmdk files), virtual floppy disks (Windows-based .vfd files or VMware-based .flp files), ISO images (.iso files), and scripts.

### Parameters

#### -AddDefaultResources

Indicates that the default resources for a library share are added.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Credential<PSCredential>**

Specifies a credential object or, for some cmdlets, a Run As account object that contains the user name and password of an account that has permission to perform this action. Or, in the case of Restart-SCJob, has permission to complete a restarted task.

For more information about the PSCredential object, type: "Get-Help Get-Credential".

For more information about Run As accounts, type: "Get-Help New-SCRunAsAccount".

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Description<String>**

States a description for the specified object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-JobGroup<Guid>**

Specifies an identifier for a series of commands that will run as a set just before the final command that includes the same job group identifier runs.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-SharePath<String>**

Specifies a path to a valid library share on an existing library server that uses a Universal Naming Convention (UNC) path.

Example format: `-SharePath "\\LibServer01\LibShare"`

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **LibraryShare**

## Examples

### 1: Add a Windows share as a library share object to the VMM library.

The first command connects to VMMServer01.

The second command adds a library share object to the library named AllVHDs (a Windows share located on LibraryServer01). This example assumes that LibraryServer01 is already a VMM library server.

```
PS C:\> Get-SCVMMServer -ComputerName "VMMServer01.Contoso.com"
PS C:\> Add-SCLibraryShare -SharePath "\\LibraryServer01\AllVHDs"
```

### 2: Add two Windows shares as library share objects to the VMM library.

The first command connects to VMMServer1.

The second command stores the strings "\\LibraryServer01\AllVHDs" and "\\LibraryServer01\AllISOs" in the \$SharePaths variable. This example assumes that LibraryServer01 is already a VMM library server.

The last command uses a foreach loop to pass the two share names stored in \$SharePaths to the Add-SCLibraryShare cmdlet, which adds each Windows share as a library share to VMM.

NOTE: For more information about the standard Windows PowerShell foreach loop statement, type: Get-Help about\_ForEach. The foreach loop statement is not the same as the Foreach-Object cmdlet, which uses "foreach" as an alias.

```
PS C:\> Get-SCVMMServer -ComputerName "VMMServer01.Contoso.com"
PS C:\> $SharePaths = "\\LibraryServer01\AllVHDs", "\\LibraryServer01\AllISOs"
PS C:\> foreach($SharePath in $SharePaths) { Add-SCLibraryShare -SharePath $SharePath }
```

## Related topics

[Find-SCLibraryShare](#)

[Get-SCDependentLibraryResource](#)

---

[Get-SCLibraryShare](#)

[Read-SCLibraryShare](#)

[Remove-SCLibraryShare](#)

[Set-SCLibraryShare](#)



---

# Add-SCLoadBalancer

---

## Add-SCLoadBalancer

Adds a load balancer to VMM.

### Syntax

Parameter Set: Default

```
Add-SCLoadBalancer [-LoadBalancerAddress] <String> -Manufacturer <String> -Model <String> -  
RunAsAccount <RunAsAccount> -VMHostGroup <HostGroup[]> [-ConfigurationProvider  
<ConfigurationProvider> ] [-JobVariable <String> ] [-LogicalNetworkDedicatedIP  
<LogicalNetwork[]> ] [-LogicalNetworkVIP <LogicalNetwork[]> ] [-Port <UInt16> ] [-PROTipID  
<Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Add-SCLoadBalancer cmdlet adds a load balancer to Virtual Machine Manager (VMM).

### Parameters

#### **-ConfigurationProvider<ConfigurationProvider>**

Specifies a configuration provider object. A configuration provider is a plug-in to VMM that translates VMM PowerShell commands to API calls that are specific to a type of load balancer. If no configuration provider is specified, VMM uses the Manufacturer and Model information to choose an available configuration provider. If no configuration provider is found, the load balancer will not be added.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-LoadBalancerAddress<String>**

Specifies the fully qualified domain name (FQDN) or IP address of a load balancer. Usual formats are FQDN, IPv4 or IPv6 addresses, but check with the load balancer manufacturer for the valid format for your load balancer.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-LogicalNetworkDedicatedIP<LogicalNetwork[]>**

Specifies the logical networks from which the back-end IP address for the load balancer should be assigned (the back-end logical network affinity).

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

### **-LogicalNetworkVIP<LogicalNetwork[]>**

Specifies the logical networks from which the front-end IP address for the load balancer should be assigned (the front-end logical network affinity).

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Manufacturer<String>**

Specifies the name of the company that manufactured a physical device.

Valid characters include: letters (a-z), numbers (0-9), underscore (\_), hyphen(-), dot(.), and single quote(').

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Model<String>**

Specifies the model of a physical device.

Aliases	none
Required?	true
Position?	named
Default Value	none

---

Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Port<UInt16>**

Specifies the network port to use when adding an object or creating a connection. Valid values are: 1 to 4095.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-RunAsAccount<RunAsAccount>**

Specifies a Run As account that contains credentials with permission to perform this action.

Aliases	none
Required?	true
Position?	named

---

Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMHostGroup<HostGroup[]>**

Specifies a virtual machine host group object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named

---

Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **LoadBalancer**

## Examples

### 1: Add a load balancer.

The first command gets the load balancer provider object with the specified manufacturer and model and stores the object in the \$LBProvider variable.

The second command creates an array named \$HostGroup. The third and fourth commands populate the \$HostGroup array with host groups named HostGroup01 and Production.

The fifth command gets the Run As account object named LBRunAsAcct and stores the object in the \$RunAsAcct variable.

The last command adds the load balancer using the specified Run As account.

```
PS C:\> $LBProvider = Get-SCConfigurationProvider | where { $_.Type -eq "LoadBalancer" -and
$_.Manufacturer -eq "LBManufacturer" -and $_.Model -eq "LB01"}
PS C:\> $HostGroup =@()
PS C:\> $HostGroup += Get-SCVMHostGroup | where { $_.Path -eq "All Hosts\HostGroup01" }
PS C:\> $HostGroup += Get-SCVMHostGroup | where { $_.Path -eq "All
Hosts\HostGroup02\Production" }
PS C:\> $RunAsAcct = Get-SCRunAsAccount -Name "LBRunAsAcct"
PS C:\> Add-SCLoadBalancer -ConfigurationProvider $LBProvider -VMHostGroup $HostGroup -
RunAsAccount $RunAsAcct -LoadBalancerAddress "LB.Contoso.com" -Manufacturer "LBManufacturer"
-Model "LB01" -Port "123"
```

## Related topics

[Get-SCLoadBalancer](#)

[Read-SCLoadBalancer](#)

[Remove-SCLoadBalancer](#)

---

[Set-SCLoadBalancer](#)

[Test-SCLoadBalancer](#)

---

# Add-SCNetworkGateway

---

## Add-SCNetworkGateway

Adds a network gateway.

### Syntax

Parameter Set: ByModel

```
Add-SCNetworkGateway [-ConnectionString] <String> -Manufacturer <String> -Model <String> -Name <String> -RunAsAccount <RunAsAccount> [-Description <String> ] [-IPv4Address <String> ] [-IPv6Address <String> ] [-JobVariable <String> ] [-LogicalNetwork <LogicalNetwork> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ByProvider

```
Add-SCNetworkGateway [-ConnectionString] <String> -ConfigurationProvider <ConfigurationProvider> -Name <String> -RunAsAccount <RunAsAccount> [-Description <String> ] [-IPv4Address <String> ] [-IPv6Address <String> ] [-JobVariable <String> ] [-LogicalNetwork <LogicalNetwork> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Add-SCNetworkGateway cmdlet adds a network Gateway to Virtual Machine Manager (VMM).

### Parameters

#### **-ConfigurationProvider<ConfigurationProvider>**

Specifies a configuration provider object. A configuration provider is a plug-in to VMM that translates VMM PowerShell commands to API calls that are specific to a type of load balancer. If no configuration provider is specified, VMM uses the Manufacturer and Model information to choose an available configuration provider. If no configuration provider is found, the load balancer will not be added.

Aliases	none
Required?	true
Position?	named
Default Value	none



---

Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-ConnectionString<String>**

Specifies the information required to connect to the virtual switch extension manager.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Description<String>**

States a description for the specified object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-IPv4Address<String>**

Specifies an IPv4 address.

Aliases	none
Required?	false
Position?	named
Default Value	none

---

Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-IPv6Address<String>**

Specifies an IPv6 address.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-LogicalNetwork<LogicalNetwork>**

Specifies a logical network. A logical network is a named grouping of IP subnets and VLANs that is used to organize and simplify network assignments.

Aliases	none
Required?	false
Position?	named

---

Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Manufacturer<String>**

Specifies the name of the company that manufactured a physical device.

Valid characters include: letters (a-z), numbers (0-9), underscore (\_), hyphen(-), dot(.), and single quote(').

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Model<String>**

Specifies the model of a physical device.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
---------	------

---

Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-RunAsAccount<RunAsAccount>**

Specifies a Run As account that contains credentials with permission to perform this action.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## **Related topics**

[Get-SCNetworkGateway](#)

[Read-SCNetworkGateway](#)

[Remove-SCNetworkGateway](#)

[Set-SCNetworkGateway](#)

[Test-SCNetworkGateway](#)

---

# Add-SCNetworkRoute

---

## Add-SCNetworkRoute

Adds a network route to VMM.

### Syntax

Parameter Set: Default

```
Add-SCNetworkRoute [-IPSubnet] <String> [-DenyRouting] [-VMMServer <ServerConnection> ] [-VPNConnection <VPNConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Add-SCNetworkRoute cmdlet adds a network route to Virtual Machine Manager (VMM).

### Parameters

#### -DenyRouting

Indicates that the specified subnet is excluded from being accessed by the network route.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

#### -IPSubnet<String>

Specifies an IP subnet object.

Aliases	none
---------	------

Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-VPNConnection<VPNConnection>**

Specifies a virtual private network (VPN) connection object.

To obtain a VPN connection object, use the Get-SCVPNConnection cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

---

## Related topics

[Get-SCVMMServer](#)

[Get-SCVPNConnection](#)



---

# Add-SCOperatingSystem

---

## Add-SCOperatingSystem

Adds an operating system to an application profile to specify which operating systems the application profile is compatible with.

### Syntax

Parameter Set: Default

```
Add-SCOperatingSystem -ApplicationProfile <ApplicationProfile> -OperatingSystem  
<OperatingSystem> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-  
VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Add-SCOperatingSystem cmdlet adds an operating system to an application profile object. Setting the operating system for an application profile determines which operating systems the profile is compatible with. If no operating system is set, by default the profile is compatible with all operating systems.

### Parameters

#### **-ApplicationProfile<ApplicationProfile>**

Specifies an application profile object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-OperatingSystem<OperatingSystem>**

Specifies the type of operating system for a virtual machine. To list the names of all available operating systems in VMM, type: "Get-SCOperatingSystem".

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **OperatingSystem**

---

## Examples

### 1: Add an operating system to an application profile.

The first command gets the application profile object named SvcWebAppProfile01 and stores the object in the \$AppProfile variable.

The second command gets the operating system object named 64-bit edition of Windows Server 2008 R2 Enterprise from VMMServer01 and stores the object in the \$OS variable.

The last command adds the operating system stores in \$OS to the application profile stored in \$AppProfile.

```
PS C:\> $AppProfile = Get-SCApplicationProfile -Name "SvcWebAppProfile01"
```

```
PS C:\> $OS = Get-SCOperatingSystem -VMMServer "VMMServer01.Contoso.com" | where {$_.Name -eq "64-bit edition of Windows Server 2008 R2 Enterprise"}
```

```
PS C:\> Add-SCOperatingSystem -ApplicationProfile $AppProfile -OperatingSystem $OS
```

## Related topics

[Remove-SCOperatingSystem](#)

---

# Add-SCPatch

---

## Add-SCPatch

Adds information about patches and binaries to the VMM patch cache.

### Syntax

Parameter Set: Default

```
Add-SCPatch [-JobVariable <String> ] [-PatchFilePath <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Add-SCPatch cmdlet adds information about patches and binaries to the Virtual Machine Manager (VMM) patch cache. Patches are required for physical-to-virtual machine conversions (P2V conversions) as well as for virtual-to-virtual machine conversions (V2V conversions).

The Add-SCPatch cmdlet updates the VMM database with information about patches and extracts required binaries to the following folder (default location): C:\Program Files\Microsoft System Center 2012\Virtual Machine Manager\VMMDData

To determine which patches are required for a particular conversion, run the appropriate cmdlet to gather information about the source:

- New-SCComputerConfiguration for P2V conversions
- New-SCVMXComputerConfiguration for V2V conversions

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false

---

Accept Wildcard Characters?	false
-----------------------------	-------

### **-PatchFilePath<String>**

Specifies the path to a folder on the file system where VMM is installed or to a network share where P2V or V2V patch files are located.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Examples**

### **1: Add a new patch from the default patch import directory.**

Before running the commands illustrated in this example, place any patch files (.cab or .exe files) into the Patch Cache folder located in the VMM installation directory on the VMM server. The default location is:

```
C:\Program Files\Microsoft System Center 2012\Virtual Machine Manager\Patch Import
```

This command extracts any patches found in the Patch Import folder and adds these patches to the VMM patch cache on VMMServer01.

NOTE: The patch files will automatically be deleted from the Patch Import folder after they are successfully added to the patch cache.

```
PS C:\> Add-SCPatch -VMMServer "VMMServer01.Contoso.com"
```

## **Related topics**

[New-SCComputerConfiguration](#)

[New-SCP2V](#)

---

[New-SCV2V](#)

[New-SCVMXComputerConfiguration](#)



---

# Add-SCPowerOptimizationRange

---

## Add-SCPowerOptimizationRange

Adds a time range to the power optimization schedule in a dynamic optimization configuration.

### Syntax

Parameter Set: FromDOSettings

```
Add-SCPowerOptimizationRange -BeginHour <Int32> -DynamicOptimizationConfiguration
<HostGroupDOSettings> -EndHour <Int32> -WeeklyScheduleDayOfWeek <Int32> [-JobVariable
<String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-Confirm] [-WhatIf] [
<CommonParameters>]
```

### Detailed Description

The Add-SCPowerOptimizationRange cmdlet adds a time range to the power optimization schedule in the dynamic optimization configuration. Power optimization is implemented only during the time ranges that have been added. Otherwise, hosts associated with the dynamic optimization configuration are turned on.

### Parameters

#### -BeginHour<Int32>

Specifies the hour of the day that power optimization begins.

Example format to begin power optimization at 3 AM: -BeginHour 3

Example format to begin power optimization at 5 PM: -BeginHour 17

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-DynamicOptimizationConfiguration<HostGroupDOSettings>**

Specifies a dynamic optimization configuration object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **-EndHour<Int32>**

Specifies the hour of the day that power optimization ends.

Example format to end power optimization at 8 AM: -EndHour 7

Example format to end power optimization at 3 PM: -EndHour 14

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-WeeklyScheduleDayOfWeek<Int32>**

Specifies one or more days of the week to run a job. The default value is the current day of the week. Valid values to specify an individual day by using a string: Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday. Valid values to specify a set of days in a week: Any set of two or more days separated by commas. Valid values to specify an individual day by using an integer: 1, 2, 3, 4, 5, 6, 7

Example format: -WeeklyScheduleDayOfWeek "Monday"

Example format: -WeeklyScheduleDayOfWeek "Monday, Wednesday, Friday"

Example format: -WeeklyScheduleDayOfWeek 5 (to specify a Friday)

Requirement: Use with StartTimeOfDay.

Aliases	none
---------	------

---

Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Confirm**

Prompts you for confirmation before executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **PowerOptimizationSchedule**

---

## Examples

### 1: Add a time range for power optimization to a dynamic optimization configuration.

The first command gets the host group object named HostGroup01 and stores the object in the \$HostGroup variable.

The second command gets the dynamic optimization configuration object for the host group stored in \$HostGroup and stores the object in the \$DOConfig variable.

The last command adds a time range to the dynamic optimization configuration stored in \$DOConfig.

```
PS C:\> $HostGroup = Get-SCVMHostGroup "HostGroup01"
```

```
PS C:\> $DOConfig = Get-SCDynamicOptimizationConfiguration -VMHostGroup $HostGroup
```

```
PS C:\> Add-SCPowerOptimizationRange -DynamicOptimizationConfiguration $DOConfig -BeginHour 19 -EndHour 23 -WeeklyScheduleDayOfWeek 0
```

## Related topics

[Get-SCDynamicOptimizationConfiguration](#)

[Get-SCPowerOptimizationRange](#)

[Get-SCVMHostGroup](#)

---

# Add-SCPXEserver

---

## Add-SCPXEserver

Adds an existing Windows Deployment Services computer to VMM.

### Syntax

Parameter Set: Default

```
Add-SCPXEserver [-ComputerName] <String> -Credential <VMMCredential> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Add-SCPXEserver cmdlet installs a Virtual Machine Manager (VMM) agent on an existing Windows Deployment Services computer and creates a PXEserver object for that computer in the VMM database.

### Parameters

#### **-ComputerName<String>**

Specifies the name of a computer that VMM can uniquely identify on your network. Valid formats are: FQDN, IPv4 or IPv6 address, or NetBIOS name.

NOTE: See the examples for a specific cmdlet to determine how that cmdlet specifies the computer name.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

---

## **-Credential<VMMSCredential>**

Specifies a credential object or, for some cmdlets, a Run As account object that contains the user name and password of an account that has permission to perform this action. Or, in the case of Restart-SCJob, has permission to complete a restarted task.

For more information about the PSCredential object, type: "Get-Help Get-Credential".

For more information about Run As accounts, type: "Get-Help New-SCRunAsAccount".

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false

---

Accept Wildcard Characters?	false
-----------------------------	-------

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **PXE Server**



---

## Examples

### 1: Add a PXE server to VMM.

The first command gets the Run As account object named RunAsAccount01 and stores the object in the \$Credential variable. The account must have local Administrator permissions on the PXE server.

The second command adds the PXE server named WDS Server01 using the Run As account stored in \$Credential.

```
PS C:\> $Credential = Get-SCRunAsAccount -Name "RunAsAcct01"
```

```
PS C:\> Add-SCPXEserver -ComputerName "WDS Server01.Contoso.com" -Credential $Credential
```

## Related topics

[Get-SCPXEserver](#)

[Remove-SCPXEserver](#)

---

# Add-SCScriptCommand

---

## Add-SCScriptCommand

Adds a script command to an application profile, application deployment, or host profile.

### Syntax

Parameter Set: ApplicationDeployment

```
Add-SCScriptCommand -ApplicationDeployment <ApplicationDeployment> -Executable <String> -ScriptType {PreService | PostService | SaveState | RestoreState | PreInstall | PostInstall | PreUninstall | PostUninstall | OrderedCommand} [-CommandParameters <String> ] [-DeploymentOrder <Int32> ] [-JobVariable <String> ] [-LibraryResource <CustomResource> ] [-PROTipID <Guid> ] [-RunAsAccount <VMMCredential> ] [-RunAsynchronously] [-ScriptCommandSetting <SCScriptCommandSetting> ] [-StandardInput <String> ] [-TimeoutSeconds <Int32> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ApplicationProfile

```
Add-SCScriptCommand -ApplicationProfile <ApplicationProfile> -Executable <String> -ScriptType {PreService | PostService | SaveState | RestoreState | PreInstall | PostInstall | PreUninstall | PostUninstall | OrderedCommand} [-CommandParameters <String> ] [-DeploymentOrder <Int32> ] [-JobVariable <String> ] [-LibraryResource <CustomResource> ] [-PROTipID <Guid> ] [-RunAsAccount <VMMCredential> ] [-RunAsynchronously] [-ScriptCommandSetting <SCScriptCommandSetting> ] [-StandardInput <String> ] [-TimeoutSeconds <Int32> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: VMHostProfile

```
Add-SCScriptCommand -Executable <String> -VMHostProfile <VMHostProfile> [-CommandParameters <String> ] [-DeploymentOrder <Int32> ] [-JobVariable <String> ] [-LibraryResource <CustomResource> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-ScriptCommandSetting <SCScriptCommandSetting> ] [-StandardInput <String> ] [-TimeoutSeconds <Int32> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Add-SCScriptCommand cmdlet adds a script command to an application profile, application deployment, or host profile. A script command allows an Administrator to run code during deployment and servicing operations.

---

## Parameters

### **-ApplicationDeployment<ApplicationDeployment>**

Specifies an application deployment object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-ApplicationProfile<ApplicationProfile>**

Specifies an application profile object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-CommandParameters<String>**

Specifies the parameters for a script or executable program.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-DeploymentOrder<Int32>**

Specifies the order in which a computer tier, application host, or application is deployed.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Executable<String>**

Specifies the name of an executable program.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-LibraryResource<CustomResource>**

Specifies a resource stored in the VMM library.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-RunAsAccount<VMMCredential>**

Specifies a Run As account that contains credentials with permission to perform this action.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-ScriptCommandSetting<SCScriptCommandSetting>**

Specifies a script command setting object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-ScriptType<ScriptCommandType>**

Specifies a script type. Valid values are: PreInstall, PostInstall, SaveState, RestoreState, PreService, PostService, PreUninstall, PostUninstall.

The acceptable values for this parameter are:

PreService	
PostService	
SaveState	
RestoreState	

---

PreInstall	
PostInstall	
PreUninstall	
PostUninstall	
OrderedCommand	

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-StandardInput<String>**

Specifies a path to a file that contains standard input information to use with the script command.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-TimeoutSeconds<Int32>**

Specifies the amount of time, in seconds, that a process waits before timing out.

Aliases	none
Required?	false
Position?	named

Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMHostProfile<VMHostProfile>**

Specifies a virtual machine host profile object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

### **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **ScriptCommand**



---

## Examples

### 1: Add a script command to an application profile.

The first command gets the application profile object named SvcWebAppProfile01 and stores the object in the \$AppProfile variable.

The second command creates a script command setting object that sets the working directory to Payload, and then stores the object in the \$ScriptSetting variable.

The last command adds a preinstall script command to the application profile stored in \$AppProfile.

```
PS C:\> $AppProfile = Get-SCApplicationProfile -Name "SvcWebAppProfile01"
```

```
PS C:\> $ScriptSetting = New-SCScriptCommandSetting -WorkingDirectory "Working_Folder_01"
```

```
PS C:\> Add-SCScriptCommand -ApplicationProfile $AppProfile -Executable "startup.ps1" -  
ScriptType "PreInstall" -ScriptCommandSetting $ScriptSetting -TimeoutSeconds 120
```

## Related topics

[Get-SCScriptCommand](#)

[Remove-SCScriptCommand](#)

[Set-SCScriptCommand](#)

---

# Add-SCServerFeature

---

## Add-SCServerFeature

Adds an operating system role or feature to a guest OS profile.

### Syntax

Parameter Set: OSProfile

```
Add-SCServerFeature -GuestOSProfile <GuestOSProfile> -ServerFeature <ServerFeature> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: Template

```
Add-SCServerFeature -ServerFeature <ServerFeature> -VMTemplate <Template> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Add-SCServerFeature cmdlet adds an operating system role or feature to a guest OS profile. The role or feature is automatically installed during machine deployment and servicing operations.

### Parameters

#### **-GuestOSProfile<GuestOSProfile>**

Specifies a guest operating system profile object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-ServerFeature<ServerFeature>**

Specifies a server feature object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **-VMTemplate<Template>**

Specifies a VMM template object used to create virtual machines.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **ServerFeature**

## Examples

### 1: Add a server feature to a guest OS profile.

The first command gets the guest OS profile object named NewOSProfile01 and stores the object in the \$OSProfile variable.

The second command gets the server feature object named Failover-Clustering and stores the object in the \$Feature variable.

The last command adds the server feature stored in \$Feature (Failover-Clustering) to the guest OS profile stored in \$OSProfile (NewOSProfile01).

```
PS C:\> $OSProfile = Get-SCGuestOSProfile -Name "NewOSProfile01"
```

```
PS C:\> $Feature = Get-SCServerFeature -Name "Failover-Clustering"
```

```
PS C:\> Add-SCServerFeature -GuestOSProfile $OSProfile -ServerFeature $Feature
```

## Related topics

[Get-SCGuestOSProfile](#)

[Get-SCServerFeature](#)

[Remove-SCServerFeature](#)

---

# Add-SCServicingWindowSubscription

---

## Add-SCServicingWindowSubscription

Adds a servicing window to a virtual machine, host, or service.

### Syntax

Parameter Set: Host

```
Add-SCServicingWindowSubscription [-ServicingWindow] <ServicingWindow> -VMHost <Host> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

Parameter Set: Service

```
Add-SCServicingWindowSubscription [-ServicingWindow] <ServicingWindow> -Service <Service> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

Parameter Set: VM

```
Add-SCServicingWindowSubscription [-ServicingWindow] <ServicingWindow> -VM <VM> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

### Detailed Description

The Add-SCServicingWindowSubscription cmdlet adds a servicing window to a virtual machine, host, or service. After a servicing window is assigned to an object, users can schedule maintenance work to be done within the servicing window by using a third-party scheduling system.

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Service<Service>**

Specifies a VMM service object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

---

## **-ServicingWindow<ServicingWindow>**

Specifies a servicing window object.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VM<VM>**

Specifies a virtual machine object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **-VMHost<Host>**

Specifies a virtual machine host object. VMM supports Hyper-V hosts, VMware ESX hosts, and Citrix XenServer hosts.

For more information about each type of host, type: "Get-Help Add-SCVMHost -detailed". See the examples for a specific cmdlet to determine how that cmdlet uses this parameter.

Aliases	none
Required?	true
Position?	named
Default Value	none



---

Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **ServiceWindowSubscription**

## Examples

### 1: Subscribe all virtual machines owned by a specific user to a servicing window.

The first command gets the servicing window object named Backup Staging A and stores the object in the \$SvcWindow variable.

The second command gets all virtual machine objects, selects only the virtual machines that are owned by Contoso\ReneeLo. and then stores those objects in the \$VMs variable.

The last command subscribes the virtual machines stored in \$VMs to the servicing window stored in \$SvcWindow.

```
PS C:\> $SvcWindow = Get-SCServiceWindow -Name "Backup Staging A"
PS C:\> $VMs = Get-SCVirtualMachine | where {$_.Owner -eq "Contoso\ReneeLo"}
PS C:\> Add-SCServiceWindowSubscription -ServiceWindow $SvcWindow -VM $VMs
```

### 2: Subscribe all virtual machines owned by a specific user to a servicing window using the pipeline operator.

The first command gets the servicing window object named Test Servers Group 3 and stores the object in the \$SvcWindow variable.

The second command gets all virtual machine objects, selects only the virtual machines that are owned by Contoso\ReneeLo and then uses the pipeline operator to pass the virtual machines to the Add-SCServiceWindowSubscription cmdlet. Add-SCServiceWindowSubscription subscribes each virtual machine that is passed to it to the servicing window stored in \$SvcWindow.

```
PS C:\> $SvcWindow = Get-SCServiceWindow -Name "Test Servers Group 3"
PS C:\> Get-SCVirtualMachine | where {$_.Owner -eq "Contoso\NevenSokec"} | Add-SCServiceWindowSubscription -ServiceWindow $SvcWindow
```

---

## Related topics

[Get-SCServicingWindowSubscription](#)

[Remove-SCServicingWindowSubscription](#)

---

# Add-SCSQLDeployment

---

## Add-SCSQLDeployment

Adds a SQL Server deployment to a SQL Server profile.

### Syntax

Parameter Set: Default

```
Add-SCSQLDeployment -AgentServiceRunAsAccount <VMMSecurityCredential> -InstanceID <String> -MediaSource <String> -Name <String> -SQLProfile <SQLProfile> -SQLServiceRunAsAccount <VMMSecurityCredential> -SQLSysAdminMemberList <String[]> [-DeploymentRunAsAccount <VMMSecurityCredential> ] [-DeploymentTimeoutSeconds <Int32> ] [-EnableNamedPipes <Boolean> ] [-EnableTCP <Boolean> ] [-InstanceName <String> ] [-JobVariable <String> ] [-MergeSQLAnswerFile <Boolean> ] [-ProductKey <String> ] [-PROTipID <Guid> ] [-ReportingServiceRunAsAccount <VMMSecurityCredential> ] [-RunAsynchronously] [-SARunAsAccount <VMMSecurityCredential> ] [-SecurityMode <String> ] [-SQLConfigurationFile <Script> ] [ <CommonParameters>]
```

### Detailed Description

The Add-SCSQLDeployment cmdlet adds a SQL Server deployment to a SQL Server profile.

### Parameters

#### **-AgentServiceRunAsAccount<VMMSecurityCredential>**

Specifies the Run As account to use for the SQL Server agent service.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-DeploymentRunAsAccount<VMMSCredential>**

Specifies the Run As account to use for installing SQL Server.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-DeploymentTimeoutSeconds<Int32>**

Specifies the amount of time, in seconds, that the SQL Server deployment waits before timing out.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-EnableNamedPipes<Boolean>**

Indicates that named pipes are used for remote connections.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

### **-EnableTCP<Boolean>**

Indicates that TCP/IP is used for remote connections.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-InstanceID<String>**

Specifies a SQL Server deployment instance ID.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-InstanceName<String>**

Specifies the SQL Server Analysis Services (SSAS) database instance name.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-MediaSource<String>**

Specifies a media source for a SQL Server deployment.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-MergeSQLAnswerFile<Boolean>**

Specifies that the cmdlet merge the specified SQL Server configuration file with the specified guest operating system settings. The default value is TRUE. This parameter is used by the VMM console. You do not need to use this parameter at the command prompt.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-ProductKey<String>**

Specifies a product key. The product key is a 25-digit number that identifies the product license. A product key can be used to register VMM or an operating system to be installed on a virtual machine or host.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false

---

Accept Wildcard Characters?	false
-----------------------------	-------

### **-ReportingServiceRunAsAccount<VMMCredential>**

Specifies the Run As account to use for Reporting Services.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-SARunAsAccount<VMMCredential>**

Specifies the Run As account to use for the SQL Server system administrator (SA) password.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false



---

Accept Wildcard Characters?	false
-----------------------------	-------

### **-SecurityMode<String>**

Specifies the security mode for SQL Server. Valid values are: WindowsAuthentication, SQLServerAuthentication.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-SQLConfigurationFile<Script>**

Specifies a SQL Server configuration file.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-SQLProfile<SQLProfile>**

Specifies a SQL Server profile object.

Aliases	none
Required?	true
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-SQLServiceRunAsAccount<VMMCredential>**

Specifies the Run As account to use for the SQL Server service.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-SQLSysAdminMemberList<String[]>**

Specifies a list of users that are SQL Server administrators.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

### **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **SQLDeployment**

---

## Examples

### 1: Add a SQL Server deployment to a SQL Server profile.

The first command gets the SQL Server profile object named SQLProfile01 and stores it in the \$SQLProfile variable.

The second command gets the RunAsProfile object named NTSystemRAP, which will be used to initiate and run the deployment, and then stores object in the \$DeploymentRunAsProfile variable.

The third command gets the RunAsProfile object named SQLAdminRAP, which defines the SA credentials for the deployment, and then stores the object in the \$SARunAsProfile variable.

The fourth command gets the RunAsProfile named NTSystemRAP, which will be used as the service account for the SQL Server and SQL Server Agent Windows services, and then stores the object in the \$SQLSvcsRunAsProfile variable.

The last command adds a SQL Server deployment to the SQLProfile01 SQL profile.

```
PS C:\> $SQLProfile = Get-SCSQLProfile -Name "SQLProfile01"
PS C:\> $DeploymentRunAsProfile = Get-SCRunAsProfile -Name "NTSystemRAP"
PS C:\> $SARunAsProfile = Get-SCRunAsProfile -Name "SQLAdminRAP"
PS C:\> $SQLSvcsRunAsProfile = Get-SCRunAsProfile -Name "NTSystemRAP"
PS C:\> Add-SCSQLDeployment -SQLProfile $SQLProfile -Name "SQL Deployment 01" -MediaSource
"C:\SQLMedia" -InstanceID "SysPrepSQL" -InstanceName "MSSQLSERVER" -DeploymentTimeoutSeconds
3600 -SQLAuthenticationType "SQLServerAuthentication" -EnableNamedPipes $True -EnableTCP
$True -SQLSysAdminMemberList @("Contoso\SQLAdmins") -ProductKey $null -
AgentServiceRunAsProfile $SQLSvcsRunAsProfile -SQLServiceRunAsProfile $SQLSvcsRunAsProfile -
DeploymentRunAsProfile $DeploymentRunAsProfile -SARunAsProfile $SARunAsProfile
```

## Related topics

[Get-SCSQLDeployment](#)

[Remove-SCSQLDeployment](#)

[Set-SCSQLDeployment](#)

---

# Add-SCSQLScriptCommand

---

## Add-SCSQLScriptCommand

Adds a SQL Server script to a SQL Server application deployment.

### Syntax

Parameter Set: Default

```
Add-SCSQLScriptCommand -ApplicationDeployment <ApplicationDeployment> -DeploymentOrder <Int32> -SQLScript <Script> -SQLScriptType {PreInstall | PostInstall | PreService | PostService | PreUninstall | PostUninstall} [-CommandParameters <String> ] [-DatabaseName <String> ] [-EncryptConnection <Boolean> ] [-ExecutionTimeoutSeconds <Int32> ] [-JobVariable <String> ] [-LoginTimeoutSeconds <Int32> ] [-OutputFilePath <String> ] [-PROTipID <Guid> ] [-RunAsAccount <VMMSCredential> ] [-RunAsynchronously] [-SQLAuthenticationType <String> ] [-WarnAndContinueOnError <Boolean> ] [ <CommonParameters>]
```

### Detailed Description

The Add-SQLScriptCommand cmdlet adds a SQL Server script to a SQL Server application deployment.

### Parameters

#### **-ApplicationDeployment<ApplicationDeployment>**

Specifies an application deployment object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

---

## **-CommandParameters<String>**

Specifies the parameters for a script or executable program.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-DatabaseName<String>**

Specifies the name of a database for a SQL Server script.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-DeploymentOrder<Int32>**

Specifies the order in which a computer tier, application host, or application is deployed.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-EncryptConnection<Boolean>**

Specifies whether the SQL Server connection is encrypted.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-ExecutionTimeoutSeconds<Int32>**

Specifies the amount of time, in seconds, that the SQL Server script command (SQL statement) will wait before timing out.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-LoginTimeoutSeconds<Int32>**

Specifies the amount of time, in seconds, that a SQL Server login waits before timing out.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-OutputFilePath<String>**

Specifies a file path to store output data from a SQL Server script.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-RunAsAccount<VMMSCredential>**

Specifies a Run As account that contains credentials with permission to perform this action.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-SQLAuthenticationType<String>**

Specifies the SQL Server authentication type. Valid values are: SQLServerAuthentication, WindowsAuthentication.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false



---

## **-SQLScript<Script>**

Specifies a SQL Server script.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-SQLScriptType<SQLScriptCommandType>**

Specifies a SQL Server script type. Valid values are: PreInstall, PostInstall, PreService, PostService, PreUninstall, PostUninstall.

The acceptable values for this parameter are:

PreInstall	
PostInstall	
PreService	
PostService	
PreUninstall	
PostUninstall	

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-WarnAndContinueOnError<Boolean>**

Indicates whether the script should warn the user and continue if the SQL Server script encounters an error while running.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **SQLScriptCommand**

## **Examples**

### **1: Add a SQL Server script to an application deployment.**

The first command gets the application profile object named SvcWebAppProfile01 and stores the object in the \$AppProfile variable.

The second command gets the application deployment object named SQLDataTierApp01 for the SvcWebAppProfile01 application profile, and then stores the object in the \$AppDeployment variable.

The third command gets the SQL Server script object named ConfigureDB.sql, release 1.0, from the VMM library and stores the object in the \$Script variable.

The last command adds the SQL Server script stored in \$Script to the application deployment stored in \$AppDeployment and sets the script type, deployment order, and database against which the script will run.

```
PS C:\> $AppProfile = Get-SCApplicationProfile -Name "SvcWebAppProfile01"
```

```
PS C:\> $AppDeployment = Get-SCApplicationDeployment -ApplicationProfile $AppProfile -Name "SQLDataTierApp01"
```

```
PS C:\> $SQLScript = Get-SCScript -Name "ConfigDB.sql" -Release "1.0"
```

```
PS C:\> Add-SCSQLScriptCommand -ApplicationDeployment $AppDeployment -SQLScriptType "PreInstall" -DeploymentOrder 1 -DatabaseName "MSSQLSERVER" -SQLScript $SQLScript
```

---

## Related topics

[Get-SCSQLScriptCommand](#)

[Remove-SCSQLScriptCommand](#)

[Set-SCSQLScriptCommand](#)

---

# Add-SCStorageProvider

---

## Add-SCStorageProvider

Adds a storage provider to VMM.

### Syntax

Parameter Set: AddSmisCimXmlProvider

```
Add-SCStorageProvider -Name <String> -NetworkDeviceName <String> -RunAsAccount <RunAsAccount> -TCPPort <UInt32> [-Certificate <ClientCertificate> ] [-Description <String> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

Parameter Set: AddSmisWmiProvider

```
Add-SCStorageProvider -AddSmisWmiProvider -ComputerName <String> -Name <String> -RunAsAccount <RunAsAccount> [-Description <String> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

Parameter Set: AddWindowsNativeWmiProvider

```
Add-SCStorageProvider -AddWindowsNativeWmiProvider -ComputerName <String> -Name <String> -RunAsAccount <RunAsAccount> [-Description <String> ] [-IsNonTrustedDomain] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

### Detailed Description

The Add-SCStorageProvider cmdlet adds a storage provider to Virtual Machine Manager (VMM) by providing the connection information required to access the provider over the network.

VMM in System Center 2012 SP1 allows you to add a WMI SMI-S provider by using the SmisWmi parameter.

### Parameters

#### -AddSmisWmiProvider

Aliases	none
Required?	true
Position?	named
Default Value	none

---

Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-AddWindowsNativeWmiProvider**

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Certificate<ClientCertificate>**

Specifies a security certificate object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-ComputerName<String>**

Specifies the name of a computer that VMM can uniquely identify on your network. Valid formats are: FQDN, IPv4 or IPv6 address, or NetBIOS name.

NOTE: See the examples for a specific cmdlet to determine how that cmdlet specifies the computer name.

Aliases	none
Required?	true

---

Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Description<String>**

States a description for the specified object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-IsNonTrustedDomain**

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
Required?	false

---

Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-NetworkDeviceName<String>**

Specifies the name of a network device.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-RunAsAccount<RunAsAccount>**

Specifies a Run As account that contains credentials with permission to perform this action.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-TCPPort<UInt32>**

Specifies a numeric value that represents a TCP port.

Aliases	none
Required?	true



---

Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **StorageProvider**

## Examples

### 1: Add a storage provider by its Fully Qualified Domain Name (FQDN).

The first command gets the RunAs account named RunAsAccount01 and stores it in the \$RunAsAcct variable.

The second command adds the storage provider named StorProv01.Contoso.com using the RunAs account stored in \$RunAsAcct.

```
PS C:\> $RunAsAcct = Get-SCRunAsAccount -Name "RunAsAccount01"
```

```
PS C:\> Add-SCStorageProvider -NetworkDeviceName "http://StorProv01.Contoso.com" -TCPPort 5988 -Name "StorProv01.Contoso.com" -RunAsAccount $RunAsAcct
```

### 2: Add a storage provider by its IP address.

The first command gets the RunAs account named RunAsAccount02 and stores it in the \$RunAsAcct variable.

The second command adds the storage provider with an IP address of 10.10.12.23 using the RunAsAccount stored in \$RunAsAcct.

```
PS C:\> $RunAsAcct = Get-SCRunAsAccount -Name "RunAsAccount02"
```

```
PS C:\> Add-SCStorageProvider -NetworkDeviceName "http://10.10.12.23" -TCPPort 5988 -Name "StorProv02.Contoso.com" -RunAsAccount $RunAsAcct02
```

---

### 3: Add a WMI SMI-S storage provider.

The first command gets the Run As account object named AdminRAA and stores the object in the \$RunAsAccount variable.

The second command adds the storage provider with the name StorageProvider01 using the Run Account stored in \$RunAsAccount. The SmisWmi parameter indicates that this is an SMI-S storage provider.

```
PS C:\> $RunAsAccount = Get-SCRunAsAccount -Name "AdminRAA"
```

```
PS C:\> Add-SCStorageProvider -SmisWmi -Name "StorageProvider01" -RunAsAccount $RunAsAccount  
-ComputerName "StorageHost01.Contoso.com"
```

### Related topics

[Get-SCStorageProvider](#)

[Read-SCStorageProvider](#)

[Remove-SCStorageProvider](#)

[Set-SCStorageProvider](#)

---

# Add-SCUpdateServer

---

## Add-SCUpdateServer

Adds a Windows Server Update Services computer to VMM.

### Syntax

Parameter Set: Default

```
Add-SCUpdateServer [-ComputerName] <String> -Credential <VMMCredential> -TCPPort <UInt32> [-  
DisableUpdateServerConfiguration] [-JobVariable <String> ] [-PROTipID <Guid> ] [-  
RunAsynchronously] [-StartUpdateServerSync] [-UseSSLConnection] [-VMMServer  
<ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Add-SCUpdateServer cmdlet adds a Microsoft Windows Server Update Services (WSUS) computer to Virtual Machine Manager (VMM). Adding a WSUS computer integrates VMM with WSUS setup and enables the update management feature.

### Parameters

#### **-ComputerName<String>**

Specifies the name of a computer that VMM can uniquely identify on your network. Valid formats are: FQDN, IPv4 or IPv6 address, or NetBIOS name.

NOTE: See the examples for a specific cmdlet to determine how that cmdlet specifies the computer name.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-Credential<VMMCredential>**

Specifies a credential object or, for some cmdlets, a Run As account object that contains the user name and password of an account that has permission to perform this action. Or, in the case of Restart-SCJob, has permission to complete a restarted task.

For more information about the PSCredential object, type: "Get-Help Get-Credential".

For more information about Run As accounts, type: "Get-Help New-SCRunAsAccount".

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-DisableUpdateServerConfiguration**

Disables the ability to control WSUS configuration settings through VMM. This parameter is used when VMM and System Center Configuration Manager are leveraging the same WSUS server and there can be only one configuration setting authority. Configuration settings are defined as the selection of production classifications, products, languages, and so on.

NOTE: By default, when you add an update server that is a downstream server managed by Configuration Manager, VMM disables its own ability to configure the WSUS settings.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
---------	------

---

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-StartUpdateServerSync**

Starts the update server synchronization process immediately after the update server is added.

Aliases	none
---------	------

---

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-TCPPort<UInt32>**

Specifies a numeric value that represents a TCP port.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-UseSSLConnection**

Indicates that the update server communicates with Windows Server Update Services (WSUS) using Secure Sockets Layer (SSL).

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

---

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **UpdateServer**

## Examples

### 1: Add an update server.

The first command gets the Run As account object named RunAsAccount01 and stores the object in the \$Credential variable.

The second command adds update server WSUSComputer01 to VMM, enabling update servicing functionality. As this command is processed, \$Credential provides Run As account credentials to Add-SCUpdateServer.

```
PS C:\> $Credential = Get-SCRunAsAccount -Name "RunAsAccount01"
```

```
PS C:\> Add-SCUpdateServer -ComputerName "WSUSComputer01" -TCPPort 80 -Credential  
$Credential
```

## Related topics

[Get-SCUpdateServer](#)

[Remove-SCUpdateServer](#)

[Set-SCUpdateServer](#)

---

# Add-SCVirtualizationManager

---

## Add-SCVirtualizationManager

Adds a VMware vCenter Server to VMM.

### Syntax

Parameter Set: Default

```
Add-SCVirtualizationManager [-ComputerName] <String> -Credential <VMMCredential> [-Certificate <ClientCertificate> ] [-Description <String> ] [-EnableSecureMode <Boolean> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-TCPPort <UInt32> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Add-SCVirtualizationManager cmdlet adds a VMware vCenter Server to your Virtual Machine Manager (VMM) environment so that VMM can connect to the vCenter Server and import its data. After you add the vCenter Server to VMM, you need to add the VMware ESX hosts associated with the vCenter Server before VMM can manage the virtual machines deployed on those hosts.

The default port used to connect to a VMware VirtualCenter Server computer is TCP port 443.

### Parameters

#### **-Certificate<ClientCertificate>**

Specifies a security certificate object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false



---

## **-ComputerName<String>**

Specifies the name of a computer that VMM can uniquely identify on your network. Valid formats are: FQDN, IPv4 or IPv6 address, or NetBIOS name.

NOTE: See the examples for a specific cmdlet to determine how that cmdlet specifies the computer name.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Credential<VMMSecurity>**

Specifies a credential object or, for some cmdlets, a Run As account object that contains the user name and password of an account that has permission to perform this action. Or, in the case of Restart-SCJob, has permission to complete a restarted task.

For more information about the PSCredential object, type: "Get-Help Get-Credential".

For more information about Run As accounts, type: "Get-Help New-SCRunAsAccount".

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **-Description<String>**

States a description for the specified object.

Aliases	none
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-EnableSecureMode<Boolean>**

Indicates whether VMM communicates with VMware ESX hosts and Citrix XenServer hosts in secure mode. The default value is \$True.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-TCPPort<UInt32>**

Specifies a numeric value that represents a TCP port.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **VirtualizationManager**

## Examples

### 1: Add a VMware vCenter Server to VMM.

The first command gets the Run As account object named RunAsAccount03 and stores the object in the \$RunAsAccount variable. The required credentials for this operation are either a local Administrator account or a domain account with administrator rights on the vCenter Server that you want to add as a virtualization manager to VMM.

The second command obtains the security certificate from vCenterServer01 and stores it in the \$Cert variable.

The last command adds the virtualization manager object named vCenterServer01 to the VMM database, imports the security certificate object, and specifies that VMM will use TCP port 443 (the default port) to connect to that server. As the last command is processed, \$Credential provides your Run As credentials to Add-SCVirtualizationManager.

```
PS C:\> $RunAsAccount = Get-SCRunAsAccount -Name "RunAsAccount03"
```

```
PS C:\> $Cert = Get-SCCertificate -Computersname "vCenterServer01.Contoso.com"
```

```
PS C:\> Add-SCVirtualizationManager -ComputerName "vCenterServer01.Contoso.com" -Certificate $Cert -TCPPort 443 -Credential $RunAsAccount
```

### 2: Add multiple VMware vCenter Servers to VMM.

The first command gets the Run As account object named RunAsAccount03 and stores the object in the \$RunAsAccount variable. The required credentials for this operation are either a local Administrator account or a domain account with administrator rights on the vCenter Server that you want to add as a virtualization manager to VMM.

---

The second command stores the strings "vCenterServer01.Contoso.com" and "vCenterServer02.Contoso.com", which are the names of two VMware vCenter Servers, in the \$Servers variable.

The last command adds the two servers to VMM and specifies that VMM will import the security certificates and use TCP port 443 (the default port) to connect to the virtualization manager service on vCenterServer01 and vCenterServer02. As this command is processed, \$Credential provides your Run As credentials to Add-SCVirtualizationManager.

For more information about the standard Windows PowerShell foreach loop statement, type: Get-Help about\_ForEach.

```
PS C:\> $Credential1 = Get-SCRunAsAccount -Name "RunAsAccount03"
```

```
PS C:\> $Servers = "vCenterServer01.Contoso.com", "vCenterServer02.Contoso.com"
```

```
PS C:\> ForEach ($Server in $Servers) {$Cert = Get-SCCertificate -Computername $Server;  
Add-SCVirtualizationManager -ComputerName $Server -Certificate $Cert -TCPport 443 -  
Credential $Credential}
```

## Related topics

[Get-SCVirtualizationManager](#)

[Read-SCVirtualizationManager](#)

[Remove-SCVirtualizationManager](#)

[Set-SCVirtualizationManager](#)

---

# Add-SCVirtualSwitchExtensionManager

---

## Add-SCVirtualSwitchExtensionManager

Adds a virtual switch extension manager to VMM.

### Syntax

Parameter Set: Default

```
Add-SCVirtualSwitchExtensionManager [-VirtualSwitchExtensionManagerConnectionString]
<String> -Manufacturer <String> -Model <String> -RunAsAccount <RunAsAccount> -VMHostGroup
<HostGroup[]> [-ConfigurationProvider <ConfigurationProvider> ] [-JobVariable <String> ] [-
PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [
<CommonParameters>]
```

### Detailed Description

The Add-SCVirtualSwitchExtensionManager cmdlet adds a virtual switch extension manager to Virtual Machine Manager (VMM).

### Parameters

#### **-ConfigurationProvider<ConfigurationProvider>**

Specifies a configuration provider object. A configuration provider is a plug-in to VMM that translates VMM PowerShell commands to API calls that are specific to a type of load balancer. If no configuration provider is specified, VMM uses the Manufacturer and Model information to choose an available configuration provider. If no configuration provider is found, the load balancer will not be added.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Manufacturer<String>**

Specifies the name of the company that manufactured a physical device.

Valid characters include: letters (a-z), numbers (0-9), underscore (\_), hyphen(-), dot(.), and single quote(').

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Model<String>**

Specifies the model of a physical device.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-RunAsAccount<RunAsAccount>**

Specifies a Run As account that contains credentials with permission to perform this action.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false



---

## **-VirtualSwitchExtensionManagerConnectionString<String>**

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMHostGroup<HostGroup[]>**

Specifies a virtual machine host group object or an array of host group objects.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Related topics

[Get-SCVirtualSwitchExtensionManager](#)

[Read-SCVirtualSwitchExtensionManager](#)

[Remove-SCVirtualSwitchExtensionManager](#)

[Set-SCVirtualSwitchExtensionManager](#)

[Test-SCVirtualSwitchExtensionManager](#)

---

# Add-SCVMHost

---

## Add-SCVMHost

Adds a computer as a virtual machine host.

### Syntax

Parameter Set: DomainJoined

```
Add-SCVMHost [-ComputerName] <String> -Credential <VMMSCredential> [-AvailableForPlacement <Boolean> ] [-CPUPercentageReserve <UInt16> ] [-Description <String> ] [-DiskSpaceReserveMB <UInt64> ] [-EnableLiveMigration <Boolean> ] [-JobVariable <String> ] [-LiveMigrationMaximum <UInt32> ] [-LiveStorageMigrationMaximum <UInt32> ] [-MaintenanceHost <Boolean> ] [-MaxDiskIOReservation <UInt64> ] [-MemoryReserveMB <UInt64> ] [-MigrationAuthProtocol {CredSSP | Kerberos} ] [-MigrationSubnet <String[]> ] [-NetworkPercentageReserve <UInt16> ] [-PROTipID <Guid> ] [-Reassociate <Boolean> ] [-RemoteConnectEnabled <Boolean> ] [-RemoteConnectPort <UInt32> ] [-RunAsynchronously] [-UseAnyMigrationSubnet <Boolean> ] [-VMHostGroup <HostGroup> ] [-VMMServer <ServerConnection> ] [-VMPaths <String> ] [ <CommonParameters> ]
```

Parameter Set: Clustered

```
Add-SCVMHost [-ComputerName] <String> -Credential <VMMSCredential> -VMHostCluster <HostCluster> [-AvailableForPlacement <Boolean> ] [-Certificate <ClientCertificate> ] [-CPUPercentageReserve <UInt16> ] [-Description <String> ] [-DiskSpaceReserveMB <UInt64> ] [-EnableLiveMigration <Boolean> ] [-EnableSecureMode <Boolean> ] [-JobVariable <String> ] [-LiveMigrationMaximum <UInt32> ] [-LiveStorageMigrationMaximum <UInt32> ] [-MaintenanceHost <Boolean> ] [-MaxDiskIOReservation <UInt64> ] [-MemoryReserveMB <UInt64> ] [-MigrationAuthProtocol {CredSSP | Kerberos} ] [-MigrationSubnet <String[]> ] [-NetworkPercentageReserve <UInt16> ] [-PROTipID <Guid> ] [-Reassociate <Boolean> ] [-RemoteConnectEnabled <Boolean> ] [-RemoteConnectPort <UInt32> ] [-RunAsynchronously] [-TCPPort <UInt32> ] [-UseAnyMigrationSubnet <Boolean> ] [-VMMServer <ServerConnection> ] [-VMPaths <String> ] [ <CommonParameters> ]
```

Parameter Set: ESXHost

```
Add-SCVMHost [-ComputerName] <String> -Credential <VMMSCredential> -VirtualizationManager <VirtualizationManager> [-AvailableForPlacement <Boolean> ] [-Certificate <ClientCertificate> ] [-CPUPercentageReserve <UInt16> ] [-Description <String> ] [-DiskSpaceReserveMB <UInt64> ] [-EnableLiveMigration <Boolean> ] [-JobVariable <String> ] [-LiveMigrationMaximum <UInt32> ] [-LiveStorageMigrationMaximum <UInt32> ] [-MaintenanceHost <Boolean> ] [-MaxDiskIOReservation <UInt64> ] [-MemoryReserveMB <UInt64> ] [-MigrationAuthProtocol {CredSSP | Kerberos} ] [-MigrationSubnet <String[]> ] [-NetworkPercentageReserve <UInt16> ] [-PROTipID <Guid> ] [-Reassociate <Boolean> ] [-RemoteConnectEnabled <Boolean> ] [-RemoteConnectPort <UInt32> ] [-RunAsynchronously] [-SshPublicKey <ClientSshPublicKey> ] [-SshPublicKeyFile <String> ] [-SshTcpPort <UInt32> ] [-TCPPort <UInt32> ] [-UseAnyMigrationSubnet <Boolean> ] [-VMHostGroup <HostGroup> ] [-VMMServer <ServerConnection> ] [-VMPaths <String> ] [ <CommonParameters> ]
```

Parameter Set: NonTrustedDomain

---

```
Add-SCVMHost [-ComputerName] <String> -Credential <VMMServerCredential> -NonTrustedDomainHost [-AvailableForPlacement <Boolean> ] [-CPUPercentageReserve <UInt16> ] [-Description <String> ] [-DiskSpaceReserveMB <UInt64> ] [-EnableLiveMigration <Boolean> ] [-JobVariable <String> ] [-LiveMigrationMaximum <UInt32> ] [-LiveStorageMigrationMaximum <UInt32> ] [-MaintenanceHost <Boolean> ] [-MaxDiskIOReservation <UInt64> ] [-MemoryReserveMB <UInt64> ] [-MigrationAuthProtocol {CredSSP | Kerberos} ] [-MigrationSubnet <String[]> ] [-NetworkPercentageReserve <UInt16> ] [-PROTipID <Guid> ] [-Reassociate <Boolean> ] [-RemoteConnectEnabled <Boolean> ] [-RemoteConnectPort <UInt32> ] [-RunAsynchronously] [-UseAnyMigrationSubnet <Boolean> ] [-VMHostGroup <HostGroup> ] [-VMMServer <ServerConnection> ] [-VMPATHS <String> ] [ <CommonParameters>]
```

Parameter Set: PerimeterNetwork

```
Add-SCVMHost [-ComputerName] <String> -EncryptionKey <PSCredential> -PerimeterNetworkHost -SecurityFile <String> [-AvailableForPlacement <Boolean> ] [-CPUPercentageReserve <UInt16> ] [-Description <String> ] [-DiskSpaceReserveMB <UInt64> ] [-EnableLiveMigration <Boolean> ] [-JobVariable <String> ] [-LiveMigrationMaximum <UInt32> ] [-LiveStorageMigrationMaximum <UInt32> ] [-MaintenanceHost <Boolean> ] [-MaxDiskIOReservation <UInt64> ] [-MemoryReserveMB <UInt64> ] [-MigrationAuthProtocol {CredSSP | Kerberos} ] [-MigrationSubnet <String[]> ] [-NetworkPercentageReserve <UInt16> ] [-PROTipID <Guid> ] [-Reassociate <Boolean> ] [-RemoteConnectEnabled <Boolean> ] [-RemoteConnectPort <UInt32> ] [-RunAsynchronously] [-UseAnyMigrationSubnet <Boolean> ] [-VMHostGroup <HostGroup> ] [-VMMServer <ServerConnection> ] [-VMPATHS <String> ] [ <CommonParameters>]
```

Parameter Set: XenServerHost

```
Add-SCVMHost [-ComputerName] <String> -Credential <VMMServerCredential> -XenServerHost [-AvailableForPlacement <Boolean> ] [-Certificate <ClientCertificate> ] [-CPUPercentageReserve <UInt16> ] [-Description <String> ] [-DiskSpaceReserveMB <UInt64> ] [-EnableLiveMigration <Boolean> ] [-EnableSecureMode <Boolean> ] [-JobVariable <String> ] [-LiveMigrationMaximum <UInt32> ] [-LiveStorageMigrationMaximum <UInt32> ] [-MaintenanceHost <Boolean> ] [-MaxDiskIOReservation <UInt64> ] [-MemoryReserveMB <UInt64> ] [-MigrationAuthProtocol {CredSSP | Kerberos} ] [-MigrationSubnet <String[]> ] [-NetworkPercentageReserve <UInt16> ] [-PROTipID <Guid> ] [-Reassociate <Boolean> ] [-RemoteConnectEnabled <Boolean> ] [-RemoteConnectPort <UInt32> ] [-RunAsynchronously] [-TCPPort <UInt32> ] [-UseAnyMigrationSubnet <Boolean> ] [-VMHostGroup <HostGroup> ] [-VMMServer <ServerConnection> ] [-VMPATHS <String> ] [ <CommonParameters>]
```

## Detailed Description

The Add-SCVMHost cmdlet adds one or more computers as virtual machine hosts to Virtual Machine Manager (VMM). A virtual machine host is a physical computer managed by VMM whose role is to host one or more virtual machines.

### TYPES OF VIRTUAL MACHINE HOSTS SUPPORTED IN SYSTEM CENTER 2012 - VMM

-----

From the perspective of networking and domains, the types of hosts that System Center 2012 - VMM supports include:

- Domain-joined Windows host - The host can be located in either a

---

trusted or untrusted domain.

- Perimeter network Windows host - A non-domain-joined Windows host can be managed in the same way as a perimeter network Windows host that is located in a domain.

- A VMware ESX host - ESX hosts do not use Windows Active Directory domains.

- A Citrix XenServer host - XenServer hosts are managed in the same way whether or not they are configured to use Windows Active Directory.

From the perspective of virtualization platform and operating system, the types of hosts that System Center 2012 - VMM supports include:

- Hyper-V hosts - A server running Windows Server 2008 or later with the Hyper-V role enabled.

- VMware ESX hosts – A VMware ESX host running proprietary software, including a hypervisor, that is managed by a VMware vCenter Server running Windows.

- Citrix XenServer hosts – A Citrix XenServer server running proprietary software, including a hypervisor.

System Center 2012 - VMM manages these three types of hosts, even though each host type implements virtualization in a different way. The following sections describe each type of host in more detail. You can review the system requirements for virtual machine hosts host in the Microsoft TechNet library at <http://go.microsoft.com/fwlink/?LinkId=217486>.

#### WHAT YOU NEED TO KNOW BEFORE YOU ADD A HYPER-V HOST

-----

- Review the system requirements for Hyper-V hosts located in the Microsoft TechNet library at <http://go.microsoft.com/fwlink/?LinkID=213749>.

- The Add-SCVMHost cmdlet will enable the Hyper-V server role for you, but you must first configure the Virtualization option in the BIOS manually.

#### WHAT YOU NEED TO KNOW BEFORE YOU ADD A WINDOWS-BASED PERIMETER NETWORK HOST

-----

To manage a Windows-based host in a perimeter network:

- Install the Virtual Machine Manager agent locally on the perimeter network host.

- When you run VMM Setup and choose the option indicating that this host is on a perimeter network, the wizard prompts you to:

- Provide an encryption key for the security file.

- Specify where you want to store the security file.

- After you have installed the local agent, obtain the security file from the folder in which it is stored. The default location is:

---

C:\Program Files\Microsoft System Center 2012\Virtual Machine Manager, and the name of the security file is SecurityFile.txt.

- Copy the security file to a location that is accessible to the computer on which a VMM console is installed.

- When you use Add-SCVMHost to add the perimeter network host, you must specify the same encryption key and point to the local security file by using the -EncryptionKey and -SecurityFile parameters. Following is an example format for these parameters:

Example format: -SecurityFile "C:\SecurityFile.txt" -EncryptionKey \$Key

Example 2 outlines the cmdlets to use to add a perimeter network host.

#### WHAT YOU NEED TO KNOW BEFORE YOU ADD A VMWARE ESX HOST

-----

- Review the system requirements for VMware ESX hosts in the Microsoft TechNet Library at <http://go.microsoft.com/fwlink/?LinkID=215885>.

#### WHAT YOU NEED TO KNOW BEFORE YOU ADD A CITRIX XENSERVEN HOST

-----

- Review the system requirements for Citrix XenServer hosts in the Microsoft TechNet library at <http://go.microsoft.com/fwlink/?LinkId=217487>.

## Parameters

### **-AvailableForPlacement<Boolean>**

Indicates whether the VMM placement process will consider this host or this volume on a host to be eligible as a possible location on which to deploy virtual machines. If this parameter is set to False, you can choose to deploy virtual machines on this host or volume anyway. The default value is True. This parameter does not apply to VMware ESX hosts.

When this parameter is used with network adapters, if set to \$False, then placement will not consider the logical networks configured on this network adapter to determine if the host is suitable for connecting a virtual machine.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-Certificate<ClientCertificate>**

Specifies a security certificate object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-ComputerName<String>**

Specifies the name of a computer that VMM can uniquely identify on your network. Valid formats are: FQDN, IPv4 or IPv6 address, or NetBIOS name.

NOTE: See the examples for a specific cmdlet to determine how that cmdlet specifies the computer name.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-CPUPercentageReserve<UInt16>**

Specifies the percentage of CPU to reserve for the use of the operating system on the physical host computer. If you do not use this parameter to specify the reserve, the default setting for the host group is used: 10 percent. The VMM placement process will not recommend that a virtual machine be placed on a host unless its resource requirements can be met without using host reserves.

Aliases	none
Required?	false
Position?	named

Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Credential<VMMSCredential>**

Specifies a credential object or, for some cmdlets, a Run As account object that contains the user name and password of an account that has permission to perform this action. Or, in the case of Restart-SCJob, has permission to complete a restarted task.

For more information about the PSCredential object, type: "Get-Help Get-Credential".

For more information about Run As accounts, type: "Get-Help New-SCRunAsAccount".

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Description<String>**

States a description for the specified object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-DiskSpaceReserveMB<UInt64>**

Specifies, in megabytes (MB), the amount of disk space to reserve for the use of the operating system on the physical host computer. If you do not use this parameter to specify the reserve, the default setting for the host group is used: 100 MB. The VMM placement process will not recommend that a



---

virtual machine be placed on a host unless its resource requirements can be met without using host reserves.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-EnableLiveMigration<Boolean>**

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-EnableSecureMode<Boolean>**

Indicates whether VMM communicates with VMware ESX hosts and Citrix XenServer hosts in secure mode. The default value is \$True.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-EncryptionKey<PSCredential>**

Specifies credentials to be used as an encryption key when you add a Hyper-V host located in a perimeter network to VMM.

Example format: -SecurityFile "C:\SecurityFile.txt" -EncryptionKey \$Key

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-LiveMigrationMaximum<UInt32>**

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-LiveStorageMigrationMaximum<UInt32>**

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-MaintenanceHost<Boolean>**

This parameter is obsolete. Use AvailableForPlacement instead.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-MaxDiskIOReservation<UInt64>**

Specifies the maximum disk I/O per second (IOPS) on the physical host computer. If you do not use this parameter to specify the reserve, the default setting for the host group is used: 10000. The VMM placement process will not recommend that a virtual machine be placed on a host unless its resource requirements can be met without using host reserves.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-MemoryReserveMB<UInt64>**

Specifies, in megabytes (MB), the amount of memory to reserve for the use of the host operating system on the physical host computer. If you do not use this parameter to specify the reserve, the default setting for the host group is used: 256 MB. The VMM placement process will not recommend that a virtual machine be placed on a host unless its resource requirements can be met without using host reserves.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-MigrationAuthProtocol<MigrationAuthProtocolType>**

The acceptable values for this parameter are:

CredSSP	
Kerberos	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-MigrationSubnet<String[]>**

Aliases	none
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-NetworkPercentageReserve<UInt16>**

Specifies the percentage of network capacity to reserve for the use of the host operating system on the physical host computer. If you do not use this parameter to specify the reserve, the default setting for the host group is used: 10 percent. The VMM placement process will not recommend that a virtual machine be placed on a host unless its resource requirements can be met without using host reserves.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-NonTrustedDomainHost**

Indicates that the host to be added to VMM belongs to a non-trusted domain.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-PerimeterNetworkHost**

Indicates that this host is located in a perimeter network.

---

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Reassociate<Boolean>**

Reassociates a host currently managed by one VMM server with another VMM server.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-RemoteConnectEnabled<Boolean>**

Enables, when set to \$True, a connection on a host server that lets users connect to their virtual machines remotely. This parameter only applies to virtual machines on Hyper-V hosts. It is not applicable to virtual machines on VMware ESX hosts or Citrix XenServer hosts.

---

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-RemoteConnectPort<UInt32>**

Specifies a default value for the TCP port to use when a remote user connects to a virtual machine. Typically, the default port for a Hyper-V host is 2179. This parameter does not apply to VMware ESX hosts or Citrix XenServer hosts.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-SecurityFile<String>**

Specifies the path to a file that contains the certificate and credentials to use for authentication of a Hyper-V host located in a perimeter network. This parameter does not apply to VMware ESX hosts or Citrix XenServer hosts.

Example format: -SecurityFile "C:\SecurityFile.txt" -EncryptionKey \$Key

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-SshPublicKey<ClientSshPublicKey>**

Specifies the public key used by Secure Shell (SSH) communications.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-SshPublicKeyFile<String>**

Specifies the path to the public key file for establishing a secured SSH channel with the target hosts.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false



---

Accept Wildcard Characters?	false
-----------------------------	-------

### **-SshTcpPort<UInt32>**

Specifies the TCP port number used by the Secure Shell (SSH) protocol.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-TCPPort<UInt32>**

Specifies a numeric value that represents a TCP port.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-UseAnyMigrationSubnet<Boolean>**

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false

---

Accept Wildcard Characters?	false
-----------------------------	-------

### **-VirtualizationManager<VirtualizationManager>**

Specifies a virtualization manager object managed by VMM.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMHostCluster<HostCluster>**

Specifies a VMM host cluster object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMHostGroup<HostGroup>**

Specifies a virtual machine host group object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false

Accept Wildcard Characters?	false
-----------------------------	-------

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-VMPaths<String>**

Specifies a set of default paths (as strings separated by the pipeline operator) on a host where virtual machine files can be stored.

Example format: -VMPaths "C:\Folder1|C:\Folder2|C:\Folder3"

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-XenServerHost**

Indicates that the specified host is a Citrix XenServer host.

Aliases	none
Required?	true
Position?	named

---

Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **Host**

## Examples

### 1: Add a host in the same domain as the VMM server.

The first command gets the Run As account object named HostComputer RunAsAccount and stores the object in the \$RunAsAccount variable. The required credentials for this operation are either a local Administrator account or a domain account with administrator rights on the computer that you want to add as a host.

The second command adds the host object named VMHost01 in the Contoso domain to VMM as a managed host, specifies a description, enables remote connections, and specifies that TCP port 5900 will be used for remote connections to VMHost01. As the last command is processed, \$RunAsAccount provides credentials to Add-SCVMHost.

```
PS C:\> $RunAsAccount = Get-SCRunAsAccount -Name "RunAsAcct01"
```

```
PS C:\> Add-SCVMHost "VMHost01.Contoso.com" -Description "This is a new host" -  
RemoteConnectEnabled $True -RemoteConnectPort 5900 -Credential $RunAsAccount
```

### 2: Add a host located in a perimeter network to VMM.

The first command prompts you for a user name and password and stores the credentials in \$Key. The user name can be any user name, but the password must be the same encryption key that was used when the VMM agent was installed on this computer. The VMM agent must be installed locally on a computer located in a perimeter network by choosing the local agent option when you run Setup. You specify the encryption key for the security file on the Security File Folder page of the System Center Virtual Machine Manager Agent Setup wizard.

The second command adds a host object that represents the computer named VMHost02 to the VMM database as a managed host. The command adds a description, disables remote connections, and specifies that this host is located in a perimeter network. This command uses the credentials stored in

---

\$Key to decrypt the contents of SecurityFile.txt (which, in this example, is located at C:\) and then uses the contents of SecurityFile.txt to authenticate the new host.

```
PS C:\> $Key = Get-SCCredential
```

```
PS C:\> Add-SCVMHost "VMHost02" -Description "This is my new perimeter network host" -  
RemoteConnectEnabled $FALSE -PerimeterNetworkHost -SecurityFile "C:\SecurityFile.txt" -  
EncryptionKey $Key
```

### 3: Add a host located in a non-trusted domain to VMM.

The first command gets the Run As account object named RunAsAccount02 and stores the object in the \$RunAsAccount variable. The required credentials for this operation are an account with administrator rights to add a host located in the non-trusted domain to the VMM server in the Contoso.com domain.

The second command adds VMHost03, located in a domain that is not trusted by Contoso.com, to the VMM database as a managed host. As this command is processed, \$RunAsAccount provides credentials to Add-SCVMHost.

```
PS C:\> $RunAsAccount = Get-SCRunAsAccount -Name "RunAsAccount02"
```

```
PS C:\> Add-SCVMHost "VMHost03.NonTrustedDomain.com" -VMMServer "VMMServer01.Contoso.com" -  
NonTrustedDomainHost -Credential $RunAsAccount
```

### 4: Add a VMware ESX host to VMM.

The first command gets the host group object named HostGroup02 and stores the object in the \$HostGroup variable.

The second command gets the Run As account object named ESX Host Computer Acct and stores the object in the \$RunAsAccount variable.

The third command gets the virtualization manager object named VirtMgrServer02 and stores the object in the \$VirtMgr variable.

The last command adds ESX Host01 to HostGroup02. The command provides the credentials in the form of a Run As account stored in \$RunAsAccount, which is required to add this host to VMM.

```
PS C:\> $HostGroup = Get-SCVMHostGroup "HostGroup02"
```

```
PS C:\> $RunAsAccount = Get-SCRunAsAccount -Name "ESX Host Computer Acct"
```

```
PS C:\> $VirtMgr = Get-SCVirtualizationManager -ComputerName "VirtMgrServer02.Contoso.com"
```

```
PS C:\> Add-SCVMHost -ComputerName "ESXHost01.Contoso.com" -Credential $RunAsAccount -  
VirtualizationManager $VirtMgr -VMHostGroup $HostGroup
```

### 5: Add a Citrix XenServer host to VMM.

The first command gets the host group object named HostGroup04 and stores the object in the \$HostGroup variable.

---

The second command gets the Run As account object named XenServer Host Computer Acct and stores the object in the \$RunAsAccount variable. The required credentials for this operation are an account with root credentials on the XenServer host.

The third command gets the certificate object for XenServerHost01 and stores the object in the \$Certificate variable.

The last command adds a XenServer as a host to HostGroup04 in VMM and provides the credentials in the form of a Run As account stored in \$RunAsAccount, which is required to add this host to VMM.

```
PS C:\> $HostGroup = Get-SCVMHostGroup "HostGroup04"
```

```
PS C:\> $RunAsAccount = Get-SCRunAsAccount -Name "XenServer Run As Acct"
```

```
PS C:\> $Certificate = Get-SCCertificate -ComputerName "XenServerHost01.Contoso.com"
```

```
PS C:\> Add-SCVMHost -ComputerName "XenServerHost01.Contoso.com" -Credential $RunAsAccount -  
VMHostGroup $HostGroup -XenServerHost -Certificate $Certificate -EnableSecureMode $True -  
TCPPort 5989
```

## Related topics

[Get-SCVMHost](#)

[Get-SCVMHostCluster](#)

[Get-SCVMHostGroup](#)

[Move-SCVMHost](#)

[New-SCVirtualMachine](#)

[Read-SCVMHost](#)

[Remove-SCVMHost](#)

[Set-SCVMHost](#)

---

# Add-SCVMHostCluster

---

## Add-SCVMHostCluster

Adds a Windows Server failover cluster, VMware ESX host cluster or Citrix XenServer resource pool to VMM.

### Syntax

Parameter Set: Default

```
Add-SCVMHostCluster [-Name] <String> -Credential <VMMCredential> [-AddVMHostJobsListVariable <String> ] [-ClusterReserve <UInt32> ] [-Description <String> ] [-EnableLiveMigration <Boolean> ] [-JobVariable <String> ] [-LiveMigrationMaximum <UInt32> ] [-LiveStorageMigrationMaximum <UInt32> ] [-MigrationAuthProtocol {CredSSP | Kerberos} ] [-MigrationSubnet <String[]> ] [-NonTrustedDomainHost] [-PROTipID <Guid> ] [-Reassociate <Boolean> ] [-RemoteConnectEnabled <Boolean> ] [-RemoteConnectPort <UInt32> ] [-RunAsynchronously] [-UseAnyMigrationSubnet <Boolean> ] [-VirtualizationManager <VirtualizationManager> ] [-VMHostGroup <HostGroup> ] [-VMMServer <ServerConnection> ] [-VMPaths <String> ] [ <CommonParameters>]
```

Parameter Set: XenServerHost

```
Add-SCVMHostCluster [-Name] <String> -Credential <VMMCredential> -XenServerHost [-AddVMHostJobsListVariable <String> ] [-Certificate <ClientCertificate[]> ] [-ClusterReserve <UInt32> ] [-Description <String> ] [-EnableLiveMigration <Boolean> ] [-EnableSecureMode <Boolean> ] [-JobVariable <String> ] [-LiveMigrationMaximum <UInt32> ] [-LiveStorageMigrationMaximum <UInt32> ] [-MigrationAuthProtocol {CredSSP | Kerberos} ] [-MigrationSubnet <String[]> ] [-PROTipID <Guid> ] [-Reassociate <Boolean> ] [-RemoteConnectEnabled <Boolean> ] [-RemoteConnectPort <UInt32> ] [-RunAsynchronously] [-TCPPort <UInt32> ] [-UseAnyMigrationSubnet <Boolean> ] [-VirtualizationManager <VirtualizationManager> ] [-VMHostGroup <HostGroup> ] [-VMMServer <ServerConnection> ] [-VMPaths <String> ] [ <CommonParameters>]
```

### Detailed Description

The Add-SCVMHostCluster cmdlet adds an existing Windows Server failover cluster, VMware ESX host cluster or a Citrix XenServer Resource Pool to the Virtual Machine Manager (VMM) database so that VMM can manage the host cluster.

Before you can use the Add-SCVMHostCluster cmdlet to add a Windows Server cluster to VMM, you must use the Failover Cluster Management tool to create and configure the host cluster. To create a host cluster by using VMM, use the Install-SCVMHostCluster cmdlet.

Before you can use the Add-SCVMHostCluster cmdlet to add a Citrix XenServer resource pool to VMM, you must use Citrix XenCenter to create and configure the resource pool.

---

Prior to using Add-SCVMHostCluster to add ESX host clusters, you must use the Add-SCVirtualizationManager cmdlet to add a VMware vCenter Server to your VMM environment and import its data. After you add the vCenter Server to VMM, you can add and manage VMware ESX clusters using VMM.

## Parameters

### **-AddVMHostJobsListVariable<String>**

Returns an array of job variable objects for jobs that are created for each node when hosts in a host cluster are added to VMM. VMM uses these job variables to track the progress of each job individually.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Certificate<ClientCertificate[]>**

Specifies a security certificate object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-ClusterReserve<UInt32>**

Specifies the number of host failures that a host cluster can sustain before VMM designates the cluster as over-committed. The default value is 1.

Aliases	none
---------	------



Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Credential<VMMCredential>**

Specifies a credential object or, for some cmdlets, a Run As account object that contains the user name and password of an account that has permission to perform this action. Or, in the case of Restart-SCJob, has permission to complete a restarted task.

For more information about the PSCredential object, type: "Get-Help Get-Credential".

For more information about Run As accounts, type: "Get-Help New-SCRunAsAccount".

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Description<String>**

States a description for the specified object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-EnableLiveMigration<Boolean>**

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-EnableSecureMode<Boolean>**

Indicates whether VMM communicates with VMware ESX hosts and Citrix XenServer hosts in secure mode. The default value is \$True.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

### **-LiveMigrationMaximum<UInt32>**

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-LiveStorageMigrationMaximum<UInt32>**

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-MigrationAuthProtocol<MigrationAuthProtocolType>**

The acceptable values for this parameter are:

CredSSP	
Kerberos	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false

---

Accept Wildcard Characters?	false
-----------------------------	-------

### **-MigrationSubnet<String[]>**

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-NonTrustedDomainHost**

Indicates that the host to be added to VMM belongs to a non-trusted domain.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false

---

Accept Wildcard Characters?	false
-----------------------------	-------

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Reassociate<Boolean>**

Reassociates a host currently managed by one VMM server with another VMM server.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-RemoteConnectEnabled<Boolean>**

Enables, when set to \$True, a connection on a host server that lets users connect to their virtual machines remotely. This parameter only applies to virtual machines on Hyper-V hosts. It is not applicable to virtual machines on VMware ESX hosts or Citrix XenServer hosts.

Aliases	none
Required?	false
Position?	named

---

Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-RemoteConnectPort<UInt32>**

Specifies a default value for the TCP port to use when a remote user connects to a virtual machine. Typically, the default port for a Hyper-V host is 2179. This parameter does not apply to VMware ESX hosts or Citrix XenServer hosts.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-TCPPort<UInt32>**

Specifies a numeric value that represents a TCP port.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-UseAnyMigrationSubnet<Boolean>**

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VirtualizationManager<VirtualizationManager>**

Specifies a virtualization manager object managed by VMM.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-VMHostGroup<HostGroup>**

Specifies a virtual machine host group object.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-VMPaths<String>**

Specifies a set of default paths (as strings separated by the pipeline operator) on a host where virtual machine files can be stored.

Example format: -VMPaths "C:\Folder1|C:\Folder2|C:\Folder3"

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-XenServerHost**

Indicates that the specified host is a Citrix XenServer host.



---

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **VMHostCluster**

## Examples

### 1: Add a failover cluster to VMM.

The first command gets the Run As account object named RunAsAccount01 and stores the object in the \$Credential variable. The Run As account used for this operation must be a domain account with administrator rights on all the nodes of the failover cluster that you want to add.

The second command gets the host group object "All Hosts". This is the host group that will be the container for the nodes in this host cluster.

The last command adds the failover cluster VMHostCluster01 to the VMM database, specifies "All Hosts" as the host group, enables remote connections, and specifies that TCP port 5900 will be used for remote connections to each node of the cluster. As the last command is processed, variable \$Credential provides the stored Run As account to Add-SCVMHostCluster.

```
PS C:\> $Credential = Get-SCRunAsAccount -Name "RunAsAccount01"
```

```
PS C:\> $VMHostGroup = Get-SCVMHostGroup | where {$_.Path -eq "All Hosts"}
```

```
PS C:\> Add-SCVMHostCluster -Name "VMHostCluster01.Contoso.com" -VMHostGroup $VMHostGroup -RemoteConnectEnabled $True -RemoteConnectPort 5900 -Credential $Credential
```

## Related topics

[Find-SCCluster](#)

[Get-SCVMHostCluster](#)

[Install-SCVMHostCluster](#)

---

[Move-SCVMHostCluster](#)

[Read-SCVMHostCluster](#)

[Remove-SCVMHostCluster](#)

[Set-SCVMHostCluster](#)

[Test-SCVMHostCluster](#)

[Uninstall-SCVMHostCluster](#)

---

# Add-SCVMHostNetworkAdapter

---

## Add-SCVMHostNetworkAdapter

Adds a physical network adapter on a host managed by VMM to a virtual network.

### Syntax

Parameter Set: Default

```
Add-SCVMHostNetworkAdapter [-VMHostNetworkAdapter] <HostNetworkAdapter> -VirtualNetwork <VirtualNetwork> [-JobGroup <Guid> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VlanEnabled] [-VlanID <UInt16> ] [-VlanMode {Access | Trunk | Invalid} ] [-VlanTrunkID <UInt16[]> ] [ <CommonParameters>]
```

### Detailed Description

The Add-SCVMHostNetworkAdapter cmdlet adds a physical network adapter (also called a network interface card, or NIC) on a host managed by Virtual Machine Manager (VMM) to a virtual network. Each virtual machine on that host can also connect through a virtual network adapter to that virtual network.

A virtual network configured on a host can connect to multiple virtual network adapters on virtual machines deployed on that host.

System Center 2012 - VMM includes virtual networking support for configuring one or more Virtual Local Area Networks (VLANs) on a host. You can use the Add-SCVMHostNetworkAdapter cmdlet or the Set-SCVMHostNetworkAdapter cmdlet to configure a single VLAN or multiple VLANs on a host. To configure corresponding VLAN settings on a virtual machine, use the New-SCVirtualNetworkAdapter cmdlet or the Set-SCVirtualNetworkAdapter cmdlet.

For an illustration of each type of VLAN, see the examples for this cmdlet.

### Parameters

#### **-JobGroup<Guid>**

Specifies an identifier for a series of commands that will run as a set just before the final command that includes the same job group identifier runs.

Aliases	none
Required?	false

---

Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VirtualNetwork<VirtualNetwork>**

Specifies a virtual network object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-VlanEnabled**

Enables a virtual LAN (VLAN) for use by virtual machines on a Hyper-V or Citrix XenServer host.

Example format for a single VLAN: -VLANEnabled -VLANMode "Access" -VLANID 35

Example format for multiple VLANs: -VLANEnabled -VLANMode "Trunk" -VLANTrunkID 1,2,100,200,1124

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VlanID<UInt16>**

Assigns a numerical identifier in the range 1-4094 to a virtual network adapter on a virtual machine or to a physical network adapter on a virtual machine host.

---

Configure a VlanID on a Hyper-V, VMware ESX, or Citrix XenServer host:

- On an externally bound physical network adapter when the Vlan mode is Access.

Configure a VlanID on a virtual network adapter of a virtual machine:

- Bound to a physical network adapter on the host, or

- Bound to an internal virtual network on the host.

Example format: -VlanEnabled

-VlanMode "Access" -VLANID 35

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VlanMode<VlanMode>**

Specifies whether a virtual LAN (VLAN) on a virtual machine host supports traffic across a single VLAN ("Access" mode) or across multiple VLANs ("Trunk" mode). Valid values are: Access, Trunk.

The acceptable values for this parameter are:

Access	
Trunk	
Invalid	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-VlanTrunkID<UInt16[]>**

Assigns a list of numerical identifiers in the range 1-4094 to a physical network adapter on a Hyper-V host.

Example format: -VLANEnabled -VLANMode "Trunk" -VLANTrunkID 1,2,100,200,1124

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMHostNetworkAdapter<HostNetworkAdapter>**

Specifies a physical network adapter object on a host to which virtual machines deployed on that host can connect.

Example format: -VMHostNetworkAdapter \$VMHostNIC

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **VMHostNetworkAdapter**

---

## Examples

### 1: Add a physical host network adapter to a virtual network.

The first command gets the host object named VMHost01 and stores the object in the \$VMHost variable.

The second command gets the virtual network object named ExternalVirtualNetwork01 on VMHost01 and stores the object in the \$VirtualNetwork variable.

The third command gets the physical network adapter object named HostAdapter01 on VMHost01 and stores the object in the \$VMHostNetworkAdapter variable.

The last command adds HostAdapter01 to ExternalVirtualNetwork01.

NOTE: You can add only one physical host adapter per virtual network. Therefore, the last command will fail if an adapter is already associated with the specified virtual network. To add a new adapter to the virtual network, you must first remove the existing host adapter.

```
PS C:\> $VMHost = Get-SCVMHost -ComputerName "VMHost01.Contoso.com"
```

```
PS C:\> $VirtualNetwork = Get-SCVirtualNetwork -VMHost $VMHost -Name  
"ExternalVirtualNetwork01"
```

```
PS C:\> $VMHostNetworkAdapter = Get-SCVMHostNetworkAdapter -VMHost $VMHost -Name  
"HostAdapter01"
```

```
PS C:\> Add-SCVMHostNetworkAdapter -VirtualNetwork $VirtualNetwork -VMHostNetworkAdapter  
$VMHostNetworkAdapter
```

### 2: Add a physical host network adapter to a VLAN that uses "Trunk" mode.

The first command gets the host object named VMHost02 and stores the object in the \$VMHost variable.

The second command gets the virtual network object named ExternalNetwork02 on VMHost02 and stores the object in the \$VirtualNetwork variable.

The third command gets the network adapter object named HostAdapter02 on VMHost02 and stores the adapter object in the \$VMHostNetworkAdapter variable.

The last command adds HostAdapter02 to virtual network ExternalNetwork02 and enables access from ExternalNetwork02 to an external networking device using 802.1Q tagged VLANs 1, 2, 100, 200, and 1124.

NOTE: You can add only one host adapter per virtual network, so the last command will fail if an adapter is already associated with the specified virtual network.

```
PS C:\> $VMHost = Get-SCVMHost -ComputerName "VMHost02"
```

```
PS C:\> $VirtualNetwork = Get-SCVirtualNetwork -VMHost $VMHost -Name "ExternalNetwork02"
```

```
PS C:\> $VMHostNetworkAdapter = Get-SCVMHostNetworkAdapter -VMHost $VMHost -Name  
"HostAdapter02"
```

```
PS C:\> Add-SCVMHostNetworkAdapter -VirtualNetwork $VirtualNetwork -VMHostNetworkAdapter  
$VMHostNetworkAdapter -VLANEnabled -VLANMode "Trunk" -VLANTrunkID 1,2,100,200,1124
```



---

### 3: Add a physical host network adapter to a VLAN that uses "Access" mode.

The first command gets the host object named VMHost03 and stores the object in the \$VMHost variable.

The second command gets the virtual network object named ExternalNetwork03 on VMHost03 and stores the object in the \$VirtualNetwork variable.

The third command gets the network adapter object named HostAdapter03 on VMHost03 and stores the adapter object in the \$VMHostNetworkAdapter variable.

The last command adds HostAdapter03 to virtual network ExternalNetwork03 and restricts access to ExternalNetwork03 to VLANID 22.

NOTE: You can add only one host adapter per virtual network, so the last command will fail if an adapter is already associated with the specified virtual network.

CAUTION: This example assumes that that your host is already connected to a VLAN or, if not, ensure that your host has two network adapters. If your host has a single network adapter, assigning the adapter to a VLAN that is unavailable to the VMM server will prevent VMM from managing the host. You can perform the steps in this example on a host that has only one network adapter if you first install the Microsoft Loopback Adapter on your server.

```
PS C:\> $VMHost = Get-SCVMHost -ComputerName "VMHost03.Contoso.com"
```

```
PS C:\> $VirtualNetwork = Get-SCVirtualNetwork -VMHost $VMHost -Name  
"ExternalVirtualNetwork03"
```

```
PS C:\> $VMHostNetworkAdapter = Get-SCVMHostNetworkAdapter -VMHost $VMHost -Name  
"HostAdapter03"
```

```
PS C:\> Add-SCVMHostNetworkAdapter -VirtualNetwork $VirtualNetwork -VMHostNetworkAdapter  
$VMHostNetworkAdapter -VLanEnabled -VLanMode "Access" -VLanID 22
```

### Related topics

[Get-SCVMHostNetworkAdapter](#)

[New-SCVirtualNetwork](#)

[New-SCVirtualNetworkAdapter](#)

[Remove-SCVMHostNetworkAdapter](#)

[Set-SCVirtualNetworkAdapter](#)

[Set-SCVMHostNetworkAdapter](#)

---

# Add-SCVMNetworkGateway

---

## Add-SCVMNetworkGateway

Adds a VM network gateway to VMM.

### Syntax

Parameter Set: Default

```
Add-SCVMNetworkGateway -Name <String> -VMNetwork <VMNetwork> [-Description <String> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RequiresVPN] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ByGateway

```
Add-SCVMNetworkGateway -Name <String> -NetworkGateway <NetworkGateway> -VMNetwork <VMNetwork> [-Description <String> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Add-SCVMNetworkGateway cmdlet adds a VM network gateway to Virtual Machine Manager (VMM).

### Parameters

#### **-Description<String>**

States a description for the specified object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-NetworkGateway<NetworkGateway>**

Specifies a network gateway object.

To obtain a network gateway object, use the Get-SCNetworkGateway cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-RequiresVPN**

Indicates that a virtual private network (VPN) is required.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **-VMNetwork<VMNetwork>**

Specifies a VM network object.

To get a VM network object, use the Get-SCVMNetwork cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## **Related topics**

[Get-SCVMNetworkGateway](#)

[Read-SCVMNetworkGateway](#)

[Remove-SCVMNetworkGateway](#)

[Set-SCVMNetworkGateway](#)

---

# Add-SCVPNConnection

---

## Add-SCVPNConnection

Adds a VPN connection to a VM network.

### Syntax

Parameter Set: Default

```
Add-SCVPNConnection [-VMNetworkGateway] <VMNetworkGateway> [-AuthenticationMethod {PSKOnly | MachineCertificates} ] [-AuthenticationTransformConstants {SHA256128 | MD596 | SHA196 | GCMAES128 | GCMAES192 | GCMAES256} ] [-Certificate <ClientCertificate> ] [-CipherTransformConstants {DES | DES3 | AES128 | AES192 | AES256 | GCMAES128 | GCMAES192 | GCMAES256} ] [-Description <String> ] [-DHGroup {None | Group1 | Group2 | Group14 | ECP256 | ECP384 | Group24} ] [-EncryptionMethod {DES | DES3 | AES128 | AES192 | AES256} ] [-IntegrityCheckMethod {MD5 | SHA1 | SHA256 | SHA384} ] [-JobVariable <String> ] [-Name <String> ] [-PFSGroup {None | PFS1 | PFS2 | PFS2048 | ECP256 | ECP384 | PFSMM | PFS24} ] [-PROTipID <Guid> ] [-Protocol {IKEv2 | L2TP | PPTP} ] [-RunAsynchronously] [-Secret <RunAsAccount> ] [-TargetIPv4VPNAddress <String> ] [-TargetIPv6VPNAddress <String> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Add-SCVPNConnection adds a virtual private network (VPN) connection to a VM network to establish a site-to-site tunnel from a Virtual Machine Manager (VMM) managed network gateway to another endpoint.

### Parameters

#### **-AuthenticationMethod<VPNAuthenticationMethod>**

Specifies a virtual private network (VPN) authentication method. Valid values are: PSKOnly, MachineCertificates.

The acceptable values for this parameter are:

PSKOnly	
MachineCertificates	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

-

## **AuthenticationTransformConstants<VPNAuthenticationTransformConstants>**

Specifies a virtual private network (VPN) transform constant. Valid values are: SHA256128, MD596, SHA196, GCMAES128, GCMAES192, GCMAES256.

The acceptable values for this parameter are:

SHA256128	
MD596	
SHA196	
GCMAES128	
GCMAES192	
GCMAES256	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Certificate<ClientCertificate>**

Specifies a security certificate object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-CipherTransformConstants<VPNCipherTransformConstants>**

Specifies a virtual private network cipher transform constants. Valid values are: DES, DES3, AES128, AES192, AES256, GCMAES128, GCMAES192, GCMAES256.

The acceptable values for this parameter are:

DES	
DES3	
AES128	
AES192	
AES256	
GCMAES128	
GCMAES192	
GCMAES256	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false



---

## **-Description<String>**

States a description for the specified object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-DHGroup<VPNDHGroup>**

The acceptable values for this parameter are:

None	
Group1	
Group2	
Group14	
ECP256	
ECP384	
Group24	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-EncryptionMethod<VPNEncryptionMethod>**

Specifies a virtual private network (VPN) encryption method. Valid values are: DES, DES3, AES128, AES192, AES256.

The acceptable values for this parameter are:

DES	
DES3	
AES128	
AES192	
AES256	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-IntegrityCheckMethod<VPNIntegrityCheckMethod>**

Specifies a virtual private network (VPN) integrity check method. Valid values are: MD5, SHA1, SHA256, SHA384.

The acceptable values for this parameter are:

MD5	
SHA1	
SHA256	
SHA384	

Aliases	none
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-PFSGroup<VPNPFSGroup>**

The acceptable values for this parameter are:

None	
PFS1	

PFS2	
PFS2048	
ECP256	
ECP384	
PFSMM	
PFS24	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

**-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

**-Protocol<VPNProtocol>**

The acceptable values for this parameter are:

IKEv2	
L2TP	

---

PPTP	
------	--

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Secret<RunAsAccount>**

Specifies a Run As account.

To obtain a Run As account, use the Get-SCRunAsAccount cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

### **-TargetIPv4VPNAddress<String>**

Specifies a target virtual private network (VPN) address in IPv4 format.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-TargetIPv6VPNAddress<String>**

Specifies a target virtual private network (VPN) address in IPv6 format.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

---

## **-VMNetworkGateway<VMNetworkGateway>**

Specifies a VM network gateway object.

To obtain a VM network gateway object, use the Get-SCVMNetworkGateway cmdlet.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

### **Related topics**

[Get-SCVPNConnection](#)

[Remove-SCVPNConnection](#)

[Set-SCVPNConnection](#)

---

# Backup-SCVMMServer

---

## Backup-SCVMMServer

Backs up the Virtual Machine Manager database.

### Syntax

Parameter Set: Default

```
Backup-SCVMMServer -Path <String> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Backup-SCVMMServer cmdlet backs up the Virtual Machine Manager (VMM) database on a VMM server to a local folder or to a remote network share. The folder to which you back up the database must be accessible to the SQL Server.

How to Determine Whether SQL Server Is Local or on a Remote Server

-----  
If you do not know whether the VMM database is stored locally or on a remote server running Microsoft SQL Server, do the following:

1. On the VMM server, open the Registry Editor.
2. Navigate to:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\  
Microsoft System Center Virtual Machine Manager Server\Settings\Sql
```

3. Look at the value for OnRemoteServer:

- If it is set to 0, the database is on the local VMM server.
- If it is set to 1, the database is on a remote SQL server.

Restoring the Backed-up Database

-----  
After you use the Backup-SCVMMServer cmdlet to back up the VMM database (see Examples 1 and 2), you can use the SCVMMRecover.exe command to restore the database (see Example 3). This command (which is not a Windows PowerShell cmdlet) is installed with VMM. By default, SCVMMRecover.exe is installed at <%system-drive%>\Program Files\Microsoft System Center 2012\Virtual Machine Manager\bin.

**IMPORTANT:** To back up and restore a server functioning as a virtual machine host or as a library server in a VMM environment, use your standard server backup and restore procedures.



---

## Parameters

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Path<String>**

Specifies the destination path for the operation.

Example formats:

Local path     -Path "F:\"

UNC path       -Path "\\Library\Templates"

Volume GUID path -Path "\\?\Volume{4703c1ea-8ae7-11db-b473-00123f7603e3}\"

VMware ESX path -Path "[storage1]\MyVMwareFolderForVMs\MyVM.vmx"

Citrix XenServer path - Path "Local storage[99b6212f-b63d-c676-25f9-d6c460992de7]"

Wildcards are supported for "Get" cmdlets and when you specify the UNC path:

Example format:

UNC path       -Path "\\VMHostServer\MyVMs\\*VM\*"

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **VMMServer**

## Examples

### 1: Back up the VMM database to a local folder.

This command backs up the VMM database on the VMM server named VMMServer01 to the specified path.

NOTE:

- Backup-SCVMMServer must back up the database to a server running Microsoft SQL Server. This example assumes that Microsoft SQL Server (for the VMM database) is installed on VMMServer01 rather than on a remote server.
- When you back up the database to a local folder, the folder must be write-accessible to the Microsoft SQL Server service.

```
PS C:\> Backup-SCVMMServer -VMMServer "VMMServer01.Contoso.com" -Path "D:\VMMBackups"
```

### 2: Back up the VMM database to a network share.

This command backs up the VMM database on the VMM server named VMMServer01 to the specified share on a server called SQLServer01.

IMPORTANT:

- \* Backup-SCVMMServer must back up the database to a server running Microsoft SQL Server, so this example assumes that Microsoft SQL Server (for the VMM database) is installed on SQLServer01.
- \* When you back up the database to a remote share, the share must be write-accessible to the Microsoft SQL Server service.

```
PS C:\> Backup-SCVMMServer -VMMServer "VMMServer01.Contoso.com" -Path "\\SQLServer01\VMMBackups"
```

### 3: Restore the VMM database.

This example demonstrates the use of SCVMMRecover.exe, and not a Windows PowerShell cmdlet. You must open a Command Prompt window (not a Windows PowerShell window) and use the

---

SCVMMRecover.exe command that is installed with VMM to perform this operation. You must run SCVMMRecover.exe locally on the VMM server on which you want to restore the database. SCVMMRecover.exe does not work with a highly available VMM installation.

This example restores the VMM database to the VMM server where:

<%backup-folder-path%> is the path on the server running Microsoft SQL Server where the .bak file is saved.

<%backup-file-name%> is the name of the .bak file that was created during the backup operation.

This example assumes that SCVMMRecover.exe is installed in the default location for VMM at:

<%system-drive%>\Program Files\Microsoft System Center 2012\Virtual Machine  
Manager\bin\SCVMMRecover.exe

C:\> SCVMMRecover.exe -Path <%backup-folder-path%>\<%backup-file-name%>.bak -Confirm

---

# Clear-SCPowerOptimizationSchedule

---

## Clear-SCPowerOptimizationSchedule

Deletes a power optimization time range from a dynamic optimization configuration.

### Syntax

Parameter Set: FromDOSettings

```
Clear-SCPowerOptimizationSchedule -DynamicOptimizationConfiguration <HostGroupDOSettings> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The Clear-SCPowerOptimizationSchedule cmdlet deletes power optimization time ranges from the power optimization schedule configured for a dynamic optimization configuration.

### Parameters

#### **-DynamicOptimizationConfiguration<HostGroupDOSettings>**

Specifies a dynamic optimization configuration object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

---

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Confirm**

Prompts you for confirmation before executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- 
- **PowerOptimizationSchedule**

## Examples

### **1: Remove all power optimization time ranges associated with a dynamic optimization configuration.**

The first command gets the host group object named HostGroup01 and stores the object in the \$HostGroup variable.

The second command gets the dynamic optimization configuration object for the host group stored in \$HostGroup and stores the object in the \$DOConfig variable.

The last command clears all time ranges in the power optimization schedule for the dynamic optimization configuration stored in \$DOConfig.

```
PS C:\> $HostGroup = Get-SCVMHostGroup "HostGroup01"
```

```
PS C:\> $DOConfig = Get-SCDynamicOptimizationConfiguration -VMHostGroup $HostGroup
```

```
PS C:\> Clear-SCPowerOptimizationSchedule -DynamicOptimizationConfiguration $DOConfig
```

## Related topics

[Get-SCDynamicOptimizationConfiguration](#)

[Get-SCVMHostGroup](#)



---

# Clear-SCPROTip

---

## Clear-SCPROTip

Dismisses a PRO tip object that is no longer applicable.

### Syntax

Parameter Set: Default

```
Clear-SCPROTip [-PROTip] <PROTip> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Clear-SCPROTip cmdlet dismisses a Performance and Resource Optimization (PRO) tip object that is no longer applicable. You can use this cmdlet to dismiss a PRO tip if, for example, its recommended action is no longer valid or is out-of-date.

Virtual Machine Manager (VMM) dismisses some PRO tips automatically if the condition that generated the alert, and the resulting PRO tip recommended action, is no longer an issue.

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

#### **-PROTip<PROTip>**

Specifies a PRO tip object.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

---

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- PROTip

## Examples

### 1: Dismiss the first active PRO tip.

The first command gets all active PRO tip objects from the VMM database and stores the objects in the \$PROTips object array.

The last command dismisses the first tip stored in \$PROTips.

```
PS C:\> $PROTips = Get-SCPROTip
```

```
PS C:\> Clear-SCPROTip -PROTip $PROTips[0]
```

## Related topics

[Get-SCPROTip](#)

[Invoke-SCPROTip](#)

[Set-SCPROTip](#)

[Test-SCPROTip](#)

---

# Compress-SCVirtualDiskDrive

---

## Compress-SCVirtualDiskDrive

Compresses a dynamically expanding virtual hard disk attached to a virtual disk drive object to reduce the size of the virtual hard disk.

### Syntax

Parameter Set: Default

```
Compress-SCVirtualDiskDrive [-VirtualDiskDrive] <VirtualDiskDrive> [-JobGroup <Guid> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

### Detailed Description

The Compress-SCVirtualDiskDrive cmdlet compresses a dynamically expanding virtual hard disk attached to a virtual disk drive object to reduce the size of the virtual hard disk. The virtual machine must be stopped before you can compress the virtual hard disk.

You can only use the Compress-SCVirtualDiskDrive cmdlet to compress a Windows-based virtual hard disk (.vhd or .vhdx) file attached to a virtual disk drive object on a virtual machine that is deployed on a Hyper-V host.

A VMware-based virtual hard disk (.vmdk) file on a virtual machine deployed on an ESX Server 3.0 or 3.5 host is fixed (not dynamic), and you cannot compress a fixed virtual hard disk.

A Citrix XenServer-based virtual hard disk (.vhd) file on a virtual machine deployed on a XenServer host is also fixed (not dynamic), and therefore cannot be compressed.

### Parameters

#### **-JobGroup<Guid>**

Specifies an identifier for a series of commands that will run as a set just before the final command that includes the same job group identifier runs.

Aliases	none
Required?	false
Position?	named
Default Value	none

---

Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VirtualDiskDrive<VirtualDiskDrive>**

Specifies a virtual disk drive object. You can attach either a virtual hard disk (for a virtual machine on any host) or a pass-through disk (for a virtual machine on a Hyper-V host or an ESX host) to a virtual disk drive object.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **VirtualDiskDrive**

## **Notes**

- Requires a VMM virtual disk drive object that is currently associated with a virtual machine deployed on a host, which can be retrieved by using the Get-SCVirtualDiskDrive cmdlet.

## **Examples**

### **1: Compress a virtual hard disk attached to a virtual disk drive on a VM deployed on a host.**

The first command gets the virtual disk drive object attached to VM01 and stores the object in the \$VDD variable. This example assumes the virtual machine has only one virtual disk drive and that the virtual hard disk attached to the virtual disk drive is, a dynamic virtual hard disk.

---

The second command compresses the dynamically expanding virtual hard disk that is attached to the virtual disk drive on VM01.

```
PS C:\> $VirtDiskDrive = Get-SCVirtualDiskDrive -VM (Get-SCVirtualMachine -Name "VM01")
```

```
PS C:\> Compress-SCVirtualDiskDrive -VirtualDiskDrive $VirtDiskDrive
```

## **Related topics**

[Convert-SCVirtualDiskDrive](#)

[Expand-SCVirtualDiskDrive](#)

[Get-SCVirtualDiskDrive](#)

[New-SCVirtualDiskDrive](#)

[Remove-SCVirtualDiskDrive](#)

[Set-SCVirtualDiskDrive](#)

---

# Convert-SCVirtualDiskDrive

---

## Convert-SCVirtualDiskDrive

Converts an existing virtual hard disk attached to a virtual disk drive object from dynamic to fixed, from fixed to dynamic, from VHD to VHDX, or converts a pass-through disk attached to a virtual disk drive object to a virtual hard disk.

### Syntax

Parameter Set: Dynamic

```
Convert-SCVirtualDiskDrive [-VirtualDiskDrive] <VirtualDiskDrive> -Dynamic [-FileName <String> ] [-JobGroup <Guid> ] [-JobVariable <String> ] [-Path <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VHDFormatType {VHD | VMDK | VHDX} ] [ <CommonParameters>]
```

Parameter Set: Fixed

```
Convert-SCVirtualDiskDrive [-VirtualDiskDrive] <VirtualDiskDrive> -Fixed [-FileName <String> ] [-JobGroup <Guid> ] [-JobVariable <String> ] [-Path <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VHDFormatType {VHD | VMDK | VHDX} ] [ <CommonParameters>]
```

Parameter Set: VHDX

```
Convert-SCVirtualDiskDrive [-VirtualDiskDrive] <VirtualDiskDrive> -VHDX [-BlockSizeBytes <Int32> ] [-DeleteSource] [-FileName <String> ] [-JobGroup <Guid> ] [-JobVariable <String> ] [-Path <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-ValidateOnCompletion] [ <CommonParameters>]
```

### Detailed Description

The Convert-SCVirtualDiskDrive cmdlet converts an existing virtual hard disk attached to a virtual disk drive object from dynamic to fixed or from fixed to dynamic, from VHD to VHDX, or converts a pass-through disk attached to a virtual disk drive object to a virtual hard disk.

CONVERTING A VIRTUAL HARD DISK FROM DYNAMIC TO FIXED, OR VICE VERSA

-----  
To convert the virtual hard disk from one format to the other, the virtual machine on which the virtual hard disk is configured must be in a stopped state.

You can only convert the disk format of a Windows-based virtual hard disk file (a .vhd file) on a virtual machine deployed on a Hyper-V host.

A VMware-based virtual hard disk file (a .vmdk file) on a virtual machine that is deployed on an ESX host is fixed in format and therefore cannot be converted to a dynamic format.

A Citrix XenServer-based virtual hard disk file (a .vhd file) on a virtual machine that is deployed on a Citrix XenServer host is also fixed in format and therefore cannot be converted to a dynamic format.



---

## CONVERTING A VHD TO VHDX

-----

To convert a VHD to VHDX, the virtual machine on which the virtual hard disk is configured must be in a stopped state.

You can only convert the disk format of a Windows-based virtual hard disk file from VHD to VHDX on a virtual machine deployed on a Hyper-V host.

## CONVERTING A PASS-THROUGH DISK TO A VIRTUAL HARD DISK

-----

A pass-through disk is a physical hard disk on the host that a virtual machine can use instead of using a virtual hard disk. You can use `Convert-SCVirtualDiskDrive` to convert a pass-through disk attached to a virtual disk drive on a virtual machine to a virtual hard disk. The virtual machine must be on a Hyper-V host, and must be in a stopped state before you can convert the pass-through disk to a virtual hard disk.

## Parameters

### **-BlockSizeBytes<Int32>**

Specifies the block size in bytes for the virtual hard disk

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-DeleteSource**

Indicates that the source virtual hard disk is deleted after a successful conversion.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-Dynamic**

Specifies that a virtual hard disk can expand dynamically.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-FileName<String>**

Specifies the file name to use when you rename a virtual hard disk file as you add it to a virtual machine.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Fixed**

Specifies that a virtual hard disk is fixed in size.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false

Accept Wildcard Characters?	false
-----------------------------	-------

### **-JobGroup<Guid>**

Specifies an identifier for a series of commands that will run as a set just before the final command that includes the same job group identifier runs.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Path<String>**

Specifies the destination path for the operation.

Example formats:

Local path     -Path "F:\"

UNC path       -Path "\\Library\Templates"

Volume GUID path -Path "\\?\Volume{4703c1ea-8ae7-11db-b473-00123f7603e3}\"

VMware ESX path -Path "[storage1]\MyVMwareFolderForVMs\MyVM.vmx"

Citrix XenServer path - Path "Local storage[99b6212f-b63d-c676-25f9-d6c460992de7]"

---

Wildcards are supported for "Get" cmdlets and when you specify the UNC path:

Example format:

UNC path      -Path "\\VMHostServer\MyVMs\\*VM\*"

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-ValidateOnCompletion**

Indicates that the converted virtual hard disk is validated after a successful conversion.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VHDFormatType<VHDFormatType>**

The acceptable values for this parameter are:

VHD	
VMDK	
VHDX	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VHDX**

Specifies that a virtual hard disk type is VHDX.

Aliases	none
Required?	true
Position?	named

Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VirtualDiskDrive<VirtualDiskDrive>**

Specifies a virtual disk drive object. You can attach either a virtual hard disk (for a virtual machine on any host) or a pass-through disk (for a virtual machine on a Hyper-V host or an ESX host) to a virtual disk drive object.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **VirtualDiskDrive**

## **Examples**

### **1: Convert a pass-through disk on a virtual disk drive on a virtual machine to a virtual hard disk.**

The first command gets the virtual machine object named VM01 and stores the object in the \$VM variable. This example assumes that VM01 is currently configured to use a pass-through disk and that the virtual machine has only one pass-through disk.

The second command gets the virtual disk drive object on VM01 and stores this object in the \$VirtDiskDrive variable.

---

The last command converts the pass-through disk drive stored in \$VirtDiskDrive to a fixed virtual hard disk and moves the virtual hard disk to the destination folder "C:\VirtualDiskDrives".

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM01"
PS C:\> $VirtDiskDrive = Get-SCVirtualDiskDrive -VM $VM
PS C:\> Convert-VirtualDiskDrive $VirtDiskDrive -Fixed -Path "C:\VirtualDiskDrives"
```

## 2: Convert one of several pass-through disks on a virtual disk drive on a virtual machine to a virtual hard disk.

The first command gets the virtual machine object named VM02 and stores the object in \$VM. This example assumes that VM02 has three virtual disk drive objects and that the first virtual disk drive is bound to a virtual hard drive whereas both the second and third virtual disk drives are bound to pass-through disks.

The second command gets all virtual disk drive objects on VM02 and stores them in the \$VirtDiskDrive object array.

The last command converts the third pass-through disk (\$VirtDiskDrive [2]) to a dynamically expanding virtual hard disk and moves this new virtual hard disk to the destination folder "D:\".

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM02"
PS C:\> $VirtDiskDrive = Get-SCVirtualDiskDrive -VM $VM
PS C:\> $VirtDiskDrive[2] | Convert-SCVirtualDiskDrive -Dynamic -Path "D:\"
```

## 3: Convert a dynamic VHD attached to a virtual disk drive object on a virtual machine to a fixed format.

The first command gets the virtual disk drive object that is attached to virtual machine VM03 and stores the virtual disk drive object in the \$VirtDiskDrive variable. This example assumes that the virtual machine has only one virtual disk drive object and that the virtual hard disk attached to the virtual disk drive is a dynamic virtual hard disk.

The second command converts the virtual hard disk stored in \$VirtDiskDrive to a fixed disk.

```
PS C:\> $VirtDiskDrive = Get-SCVirtualDiskDrive -VM (Get-SCVirtualMachine -Name "VM03")
PS C:\> Convert-SCVirtualDiskDrive -VirtualDiskDrive $VirtDiskDrive -Fixed
```

## 4: Convert a dynamic VHD attached to a virtual disk drive object on a virtual machine to fixed VHDX.

The first command gets the virtual disk drive object that is attached to virtual machine VM03 and stores the object in the \$VirtDiskDrive variable. This example assumes that the virtual machine has only one virtual disk drive object and that the virtual hard disk attached to the virtual disk drive is a dynamic virtual hard disk. This example also assumes that the virtual machine is in a stopped state.

The second command converts the virtual hard disk stored in \$VirtDiskDrive to a VHDX disk.

---

```
PS C:\> $VirtDiskDrive = Get-SCVirtualDiskDrive -VM (Get-SCVirtualMachine -Name "VM03")
PS C:\> Convert-SCVirtualDiskDrive -VirtualDiskDrive $VirtDiskDrive -VHDX
```

## 5: Convert a differencing VHD attached to a virtual disk drive on a virtual machine to a differencing VHDX.

The first command gets the virtual disk drive object that is attached to virtual machine VM04 and stores the object in the \$VirtDiskDrive variable. This example assumes that the virtual machine has only one virtual disk drive object and that the virtual hard disk attached to the virtual disk drive is a dynamic virtual hard disk. This example also assumes that the virtual machine is in a stopped state.

The second command converts the differencing virtual hard disk stored in \$VirtDiskDrive to a differencing VHDX disk. Additionally, this command deletes the source disk.

```
PS C:\> $VirtDiskDrive = Get-VirtualDiskDrive -VM (Get-SCVirtualMachine -Name "VM04")
PS C:\> Convert-VirtualDiskDrive -VirtualDiskDrive $VirtDiskDrive -VHDX -BlockSizeBytes 1MB
-DeleteSource -ValidateOnCompletion
```

## Related topics

[Compress-SCVirtualDiskDrive](#)

[Expand-SCVirtualDiskDrive](#)

[Get-SCVirtualDiskDrive](#)

[New-SCVirtualDiskDrive](#)

[Remove-SCVirtualDiskDrive](#)

[Set-SCVirtualDiskDrive](#)



---

# Copy-SCStorageVolume

---

## Copy-SCStorageVolume

Copies a volume of a physical hard disk on a source computer to a Windows-based virtual hard disk file (a .vhd file) on the specified VMM host.

### Syntax

Parameter Set: Install

```
Copy-SCStorageVolume -SourceComputerName <String> -VMHost <Host> [-AddDiskSizeMB <UInt64> ]  
[-Credential <PSCredential> ] [-DriverPath <String> ] [-Dynamic] [-Fixed] [-JobGroup <Guid>  
] [-JobVariable <String> ] [-Offline] [-Owner <String> ] [-Path <String> ] [-PROTipID <Guid>  
] [-RunAsynchronously] [-Shutdown] [-Trigger] [-VMMServer <ServerConnection> ] [-  
VolumeDeviceID <String> ] [ <CommonParameters>]
```

Parameter Set: NoInstall

```
Copy-SCStorageVolume -ComputerConfiguration <MachineConfiguration> -VMHost <Host> [-  
AddDiskSizeMB <UInt64> ] [-Credential <PSCredential> ] [-DriverPath <String> ] [-Dynamic] [-  
Fixed] [-JobGroup <Guid> ] [-JobVariable <String> ] [-Offline] [-Owner <String> ] [-Path  
<String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-Shutdown] [-Trigger] [-VMMServer  
<ServerConnection> ] [-VolumeDeviceID <String> ] [ <CommonParameters>]
```

### Detailed Description

The Copy-SCStorageVolume cmdlet copies a volume of a physical hard disk on a source computer to a Windows-based virtual hard disk file (a .vhd file) on the specified Virtual Machine Manager (VMM) host. If the volume contains an operating system, after you run Copy-SCStorageVolume, you must use the New-SCP2V cmdlet to configure the operating system to run in a virtual environment.

When you use Copy-SCStorageVolume with the -Offline parameter, the computer whose hard disk is to be copied is first started in the Windows Preinstallation Environment (Windows PE) and then the volumes are copied.

### Parameters

#### **-AddDiskSizeMB<UInt64>**

Specifies, in megabytes (MB), the amount of additional disk space to add to a virtual hard disk when performing a physical-to-virtual (P2V) or virtual-to-virtual (V2V) machine conversion. Volumes located on the virtual hard disk are automatically extended to fill the entire virtual hard disk.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-ComputerConfiguration<MachineConfiguration>**

Specifies a computer configuration object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **-Credential<PSCredential>**

Specifies a credential object or, for some cmdlets, a Run As account object that contains the user name and password of an account that has permission to perform this action. Or, in the case of Restart-SCJob, has permission to complete a restarted task.

For more information about the PSCredential object, type: "Get-Help Get-Credential".

For more information about Run As accounts, type: "Get-Help New-SCRunAsAccount".

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

---

## **-DriverPath<String>**

Specifies the path to drivers for an offline physical-to-virtual machine conversion (P2V conversion).

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Dynamic**

Specifies that a virtual hard disk can expand dynamically.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Fixed**

Specifies that a virtual hard disk is fixed in size.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-JobGroup<Guid>**

Specifies an identifier for a series of commands that will run as a set just before the final command that includes the same job group identifier runs.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Offline**

Specifies that the operation is performed offline.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-Owner<String>**

Specifies the owner of a VMM object in the form of a valid domain user account.

Example format: -Owner "Contoso\ReneeLo"

Example format: -Owner "ReneeLo@Contoso"

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Path<String>**

Specifies the destination path for the operation.

Example formats:

Local path -Path "F:\"

UNC path -Path "\\Library\Templates"

Volume GUID path -Path "\\?\Volume{4703c1ea-8ae7-11db-b473-00123f7603e3}\"

VMware ESX path -Path "[storage1]\MyVMwareFolderForVMs\MyVM.vmx"

Citrix XenServer path - Path "Local storage[99b6212f-b63d-c676-25f9-d6c460992de7]"

Wildcards are supported for "Get" cmdlets and when you specify the UNC path:

Example format:

UNC path -Path "\\VMHostServer\MyVMs\\*VM\*"

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

---

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Shutdown**

Indicates that a virtual machine, service, or a source server should be shut down. In the case of a virtual machine or service, the associated cmdlet attempts to use the operating system to shut the virtual machine down gracefully. In the case of a successful physical-to-virtual machine (P2V) conversion, the cmdlet shuts down the source server.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-SourceComputerName<String>**

Specifies the source computer for a physical-to-virtual (P2V) machine conversion performed by VMM. Valid formats: FQDN, IPv4 or IPv6 address, or NetBIOS name.

Note: See the examples for a specific cmdlet to determine how that cmdlet specifies the source computer name.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Trigger**

Starts running the commands in a job group for a physical-to-virtual (P2V) conversion, a virtual-to-virtual (V2V) conversion, or the conversion of a physical hard disk to a virtual hard disk.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMHost<Host>**

Specifies a virtual machine host object. VMM supports Hyper-V hosts, VMware ESX hosts, and Citrix XenServer hosts.

For more information about each type of host, type: "Get-Help Add-SCVMHost -detailed". See the examples for a specific cmdlet to determine how that cmdlet uses this parameter.

Aliases	none
Required?	true
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-VolumeDeviceID<String>**

Specifies the device ID of the volume to convert in a physical-to-virtual machine conversion (P2V conversion).

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

### **Outputs**

The output type is the type of the objects that the cmdlet emits.



- 
- **StorageVolume**

## Notes

- Requires a VMM computer configuration object, which can be retrieved by using the Get-SCComputerConfiguration cmdlet.

## Examples

### 1: Copy a physical hard disk from a source machine to a virtual hard disk file.

The first command uses Get-Credential to prompt you to supply a user name and password and stores your credentials in variable \$Credential. The required credentials for this operation are either a local Administrator account or a domain account with administrator rights on the computer on which resides the physical hard disk that you want to convert to a virtual hard disk.

The second command gets the host object named VMHost01 and stores the object in the \$VMHost variable.

The last command copies and converts the "C:" volume located on the source computer named P2VSource01.Contoso.com into a new virtual hard disk (still named "C:") and places it on VMHost01 at the specified path (D:\VHDs). The Fixed parameter specifies that the .vhd is fixed rather than dynamic in format, and the DiskSizeAdd parameter increases the size of the volume by 1024 MB. As this command is processed, \$Credential provides your credentials to Copy-SCStorageVolume.

```
PS C:\> $Credential = Get-Credential
```

```
PS C:\> $VMHost = Get-SCVMHost -ComputerName "VMHost01.Contoso.com"
```

```
PS C:\> Copy-SCStorageVolume -SourceComputerName "P2VSource01.Contoso.com" -VolumeDeviceID "C" -Credential $Credential -VMHost $VMHost -Path "D:\VHDs" -Fixed -DiskSizeAdd 1024
```

### 2: Copy a physical hard disk and configure the operating system on that volume to run in a virtual environment.

The first command uses Get-Credential to prompt you to supply a user name and password and stores your credentials in variable \$Credential. The required credentials for this operation are either a local Administrator account or a domain account with administrator rights on the computer on which resides the physical hard disk that you want to convert to a virtual hard disk.

The second command gets the host object named VMHost01 and stores the object in the \$VMHost variable.

The third command gathers the computer configuration information from the physical source machine called P2VSource02.Contoso.com and stores the computer configuration information in \$ComputerConfig. As this command is processed, \$Credential provides your credentials to New-SCComputerConfiguration.

The fourth command copies and converts the "C:" volume located on the source computer named P2VSource02 into a new virtual hard disk (still named "C:") and places it on VMHost01 at the specified

---

path (D:\VMs). The Fixed parameter specifies that the .vhd is fixed rather than dynamic in format, and the DiskSizeAdd parameter increases the size of the volume by 1024 MB. As this command is processed, \$Credential provides your credentials to Copy-SCStorageVolume.

The last command uses the virtual hard disk located at D:\VMs on VMHost01 and the machine configuration stored in \$ComputerConfig to create a new virtual machine called VM01. The New-SCP2V cmdlet automatically configures the operating system on the virtual hard disk to run in a virtual environment; it uses the -MemoryMB parameter to assign 512 MB of memory on the host for use by the virtual machine; and it uses the RunAsynchronously parameter to return control to the shell immediately, before the command completes.

```
PS C:\> $Credential = Get-Credential
```

```
PS C:\> $VMHost = Get-VMHost -ComputerName "VMHost01.Contoso.com"
```

```
PS C:\> $ComputerConfig = New-SCComputerConfiguration -SourceComputerName  
"P2VSource.Contoso.com" -Credential $Credential
```

```
PS C:\> Copy-SCStorageVolume -SourceComputerName "P2VSource02.Contoso.com" -VolumeDeviceID  
"C" -Credential $Credential -VMHost $VMHost -Path "D:\VMs" -Fixed -DiskSizeAdd 1024
```

```
PS C:\> New-P2V -ComputerConfiguration $ComputerConfig -Name "VM01" -VMHost $VMHost -Path  
"D:\VMs" -MemoryMB 512 -Credential $Credential -RunAsynchronously
```

## Related topics

[Get-SCComputerConfiguration](#)

[Get-SCVMMServer](#)

[New-SCComputerConfiguration](#)

[New-SCP2V](#)

---

# Copy-SCVirtualHardDisk

---

## Copy-SCVirtualHardDisk

Copies a VMware virtual hard disk file (a .vmdk file) to a Windows-based virtual hard disk file (a .vhd file) and converts the virtual hard disk for use in a VMM environment.

### Syntax

Parameter Set: Default

```
Copy-SCVirtualHardDisk -Path <String> -VMDKPath <String> -VMHost <Host> [-JobVariable <String> ] [-LibraryServer <LibraryServer> ] [-Owner <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-SourceVMHost <Host> ] [-VMMServer <ServerConnection> ] [ <CommonParameters> ]
```

### Detailed Description

The Copy-SCVirtualHarkDisk cmdlet copies a VMware virtual hard disk file (a .vmdk file) to a Windows-based virtual hard disk file (a .vhd file) and converts the virtual hard disk for use in a Virtual Machine Manager (VMM) environment. The disk's contents are preserved by this copy operation.

VMware virtual hard disks, stored in .vmdk files, contain the virtual machine's guest operating system, applications, and data. VMWare virtual hard disk formats supported by Copy-SCVirtualHardDisk include:

- monolithicSparse
- monolithicFlat
- vmfs
- twoGbMaxExtentSparse
- twoGbMaxExtentFlat

The Copy-SCVirtualHardDisk cmdlet takes as its input the .vmdk file that the .vmx file points to:

- The .vmx file points to a .vmdk file that contains metadata, which in turn points to the binary .vmdk file.
- The .vmdk file that you specify with the Copy-VMDK cmdlet is the .vmdk file that contains the metadata (not the binary .vmdk file).

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-LibraryServer<LibraryServer>**

Specifies a VMM library server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Owner<String>**

Specifies the owner of a VMM object in the form of a valid domain user account.

Example format: -Owner "Contoso\ReneeLo"

Example format: -Owner "ReneeLo@Contoso"

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-Path<String>**

Specifies the destination path for the operation.

Example formats:

Local path -Path "F:\"

UNC path -Path "\\Library\Templates"

Volume GUID path -Path "\\?\Volume{4703c1ea-8ae7-11db-b473-00123f7603e3}\"

VMware ESX path -Path "[storage1]\MyVMwareFolderForVMs\MyVM.vmx"

Citrix XenServer path - Path "Local storage[99b6212f-b63d-c676-25f9-d6c460992de7]"

Wildcards are supported for "Get" cmdlets and when you specify the UNC path:

Example format:

UNC path -Path "\\VMHostServer\MyVMs\\*VM\*"

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-SourceVMHost<Host>**

Specifies the source virtual machine host object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMDKPath<String>**

Specifies the path to a VMware virtual hard disk file (a .vmdk file) to be converted to a Windows-based virtual hard disk file (a .vhd file). The VMDK stands for the Virtual Machine Disk (VMDK) file format.

Example format: -VMDKPath "\\FileServer01\MSSCVMMLibrary\VMDKS\VM01.vmdk"

Example format: -VMDKPath "[storage1] /VM01/VM01.vmdk"

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-VMHost<Host>**

Specifies a virtual machine host object. VMM supports Hyper-V hosts, VMware ESX hosts, and Citrix XenServer hosts.

For more information about each type of host, type: "Get-Help Add-SCVMHost -detailed". See the examples for a specific cmdlet to determine how that cmdlet uses this parameter.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **StandaloneVirtualHardDisk[]**

---

## Examples

### 1: Convert a VMware .vmdk file in the VMM library to a Windows-based .vhd file on a host.

The first command gets the library server object named LibServer01 and stores the object in the \$LibServ variable.

The second command gets the host object named VMHost01 and stores the object in the \$VMHost variable. VMHost01 is a Hyper-V host.

The last command copies and converts the .vmdk file located at the specified path (\\LibServer01\MSSCVMMLibrary\VMware\VM01.vmdk) on the library server and stores the resulting .vhd file at the specified path (C:\StoredWindowsVMs) on VMHost01. Note that the Path parameter, when used with Copy-SCVirtualHardDisk, cannot take a UNC path.

Note: Copy-SCVirtualHardDisk takes as its input the .vmdk file that the .vmx file points to:

- The .vmx file points to a .vmdk file that contains metadata, which in turn points to the binary .vmdk file.
- The .vmdk file that you specify with Copy-SCVirtualHardDisk is the .vmdk file that contains the metadata (not the binary .vmdk file).

```
PS C:\> $LibServ = Get-SCLibraryServer -ComputerName "LibServer01.Contoso.com"
```

```
PS C:\> $VMHost = Get-SCVMHost -ComputerName "VMHost01.Contoso.com"
```

```
PS C:\> Copy-SCVirtualHardDisk -LibraryServer $LibServ -VMDKPath  
"\\LibServer01\MSSCVMMLibrary\VMware\VM01.vmdk" -VMHost $VMHost -Path "C:\StoredWindowsVMs"
```

## Related topics

[Get-SCVMMServer](#)

[New-SCV2V](#)

[New-SCVMXComputerConfiguration](#)



---

# Disable-SCLoadBalancerVIPMember

---

## Disable-SCLoadBalancerVIPMember

Disables a member of a load balancer VIP.

### Syntax

Parameter Set: SingleInstance

```
Disable-SCLoadBalancerVIPMember [-LoadBalancerVIPMember] <LoadBalancerVIPMember> [-  
JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer  
<ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Disable-SCLoadBalancerVIPMember cmdlet disables a member of a load balancer virtual IP (VIP), and places the member in a Desired Status of Down.

To enable a load balancer VIP member, use the Enable-SCLoadBalancerVIPMember cmdlet.

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

#### **-LoadBalancerVIPMember<LoadBalancerVIPMember>**

Specifies a member of a virtual IP (VIP) in a load balancer.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

---

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **LoadBalancerVIPMember**

## Examples

### 1: Disable a member of a load balancer virtual IP (VIP).

The first command gets the load balancer object with the address of LB01.Contoso.com and stores the object in the \$LoadBalancer variable.

The second command gets the load balancer VIP with the IP address 192.168.0.1 for the load balancer stored in \$LoadBalancer and stores the object in the \$VIP variable.

The third command gets the VIP member for the load balancer VIP stored in \$VIP with the address of 192.168.0.1 and stores the object in the \$VIPMember variable.

The last command disables the VIP member stored in \$VIPMember and displays information about the VIP member to the user.

```
PS C:\> $LoadBalancer = Get-SCLoadBalancer -LoadBalancerAddress "LB01.Contoso.com"
PS C:\> $VIP = Get-SCLoadBalancerVIP -LoadBalancer $LoadBalancer -IPAddress "192.168.0.1"
PS C:\> $VIPMember = Get-SCLoadBalancerVIPMember -LoadBalancerVIP $VIP -IPAddress
"192.168.0.1"
PS C:\> Disable-SCLoadBalancerVIPMember -LoadBalancerVIPMember $VIPMember
```

## Related topics

[Enable-SCLoadBalancerVIPMember](#)

[Get-SCLoadBalancerVIP](#)

---

[Get-SCLoadBalancerVIPMember](#)

[New-SCLoadBalancerVIPMember](#)

[Remove-SCLoadBalancerVIPMember](#)

---

# Disable-SCRunAsAccount

---

## Disable-SCRunAsAccount

Disables a Run As account so that it cannot be used.

### Syntax

Parameter Set: RunAsAccount

```
Disable-SCRunAsAccount [-RunAsAccount] <RunAsAccount> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: WithJobGroup

```
Disable-SCRunAsAccount -JobGroup <Guid> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Disable-SCRunAs cmdlet disables a Run As account so that it cannot be used by Virtual Machine Manger (VMM). To re-enable the Run As account, use the Enable-SCRunAsAccount cmdlet.

### Parameters

#### **-JobGroup<Guid>**

Specifies an identifier for a series of commands that will run as a set just before the final command that includes the same job group identifier runs.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-RunAsAccount<RunAsAccount>**

Specifies a Run As account that contains credentials with permission to perform this action.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

---

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **RunAsAccount**

---

## Examples

### 1: Disable a Run As account.

The first command gets the Run As account object named RunAsAccount01 and stores the object in the \$RunAsAccount variable.

The second command disables the Run As account stored in \$RunAsAccount.

```
PS C:\> $RunAsAccount = Get-SCRunAsAccount -Name "RunAsAccount01"
```

```
PS C:\> Disable-SCRunAsAccount -RunAsAccount $RunAsAccount
```

## Related topics

[Get-SCRunAsAccount](#)

[Remove-SCRunAsAccount](#)

[Enable-SCRunAsAccount](#)



---

# Disable-SCVMHost

---

## Disable-SCVMHost

Places a virtual machine host into maintenance mode.

### Syntax

Parameter Set: NoVMMigration

```
Disable-SCVMHost [-VMHost] <Host> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

Parameter Set: InCluster

```
Disable-SCVMHost [-VMHost] <Host> [-JobVariable <String> ] [-MoveWithinCluster] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

### Detailed Description

The Disable-SCVMHost cmdlet places a virtual machine host that is managed by Virtual Machine Manager (VMM) into maintenance mode so that you can perform maintenance tasks on the host such as applying security updates or replacing hardware.

You can use the Disable-SCVMHost cmdlet to put individual Hyper-V hosts, VMware ESX hosts, or Citrix XenServer hosts into maintenance mode. You can also use Disable-SCVMHost to put clustered hosts into maintenance mode.

To return the host to service, use the Enable-SCVMHost cmdlet.

#### HOW TO PUT HYPER-V HOSTS INTO MAINTENANCE MODE

If the host belongs to a cluster that supports live migration, you can choose either of the following methods:

**Method 1: Migrate highly available virtual machines and save the other virtual machines.** When the MoveWithinCluster parameter is used with Disable-SCVMHost, the cmdlet uses Live Migration to migrate all running highly available virtual machines to other hosts in the cluster. It places the running virtual machines that are not highly available into a saved state, which causes users to lose service. Then, it places the host into maintenance mode.

**Method 2: Save all the virtual machines.** Disable-SCVMHost places all the running virtual machines into a saved state, which causes users to lose service. Then, it places the host into maintenance mode.

If the host does not belong to a cluster, or if it belongs to a cluster that does not support live migration, Disable-SCVMHost places all of the running virtual machines into a saved state, which causes users to lose service. Then, it places the host into maintenance mode.

#### HOW DISABLE-SCVMHOST WORKS WITH VMWARE ESX HOSTS

---

When you place an ESX host into maintenance mode using `Disable-SCVMHost`, VMM sends a request to enter maintenance mode to the VMware vCenter Server that manages that host. The vCenter Server places the ESX host into maintenance mode.

NOTE: The system behavior of the virtual machines on the ESX Server host is determined by the configuration of the vCenter Server. For example, if the VMware Distributed Resources Scheduler is not configured, you might have to manually shut down all the virtual machines on the host. Or, you might have to move the virtual machines to another host to successfully place the ESX Server host into maintenance mode.

#### HOW TO PUT XENSERVER HOSTS INTO MAINTENANCE MODE

If the host belongs to a cluster that supports live migration, you can choose either of the following methods:

Method 1: Migrate highly available virtual machines and save the other virtual machines. When the `MoveWithinCluster` parameter is used with `Disable-SCVMHost`, the cmdlet uses XenServer Live Migration to migrate all running highly available virtual machines to other hosts in the cluster. It also places all the running virtual machines that are not highly available into a saved state, which causes users to lose service. Then, it places the host into maintenance mode.

Method 2: Save all the virtual machines. `Disable-VMHost` places all the running virtual machines into a saved state, which causes users to lose service. Then, it places the host into maintenance mode.

If the host does not belong to a cluster, or if it belongs to a cluster that does not support live migration, `Disable-VMHost` places all of the running virtual machines into a saved state, which causes users to lose service. Then, it places the host into maintenance mode.

#### HOST BEHAVIOR IN MAINTENANCE MODE

After you place a host into maintenance mode, the following actions are affected:

- Virtual machines cannot be created on the host.
- Virtual machines cannot be migrated to the host.
- The host is excluded from host ratings calculations performed during virtual machine placement.
- The host status is not updated.

However, you can perform the following actions:

- Remove the host from VMM if you make sure that the host is available and that its agent is in an appropriate state.
- Start or stop virtual machines on the host.
- Change the host properties.
- Migrate a virtual machine from the host to another host.

## Parameters

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-MoveWithinCluster**

Indicates that all virtual machines currently deployed on a host that is a member of a host cluster will be migrated to another host in the same host cluster if that host is placed into maintenance mode.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMHost<Host>**

Specifies a virtual machine host object. VMM supports Hyper-V hosts, VMware ESX hosts, and Citrix XenServer hosts.

For more information about each type of host, type: "Get-Help Add-SCVMHost -detailed". See the examples for a specific cmdlet to determine how that cmdlet uses this parameter.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **Host**
- **Host[]**

---

## Notes

- A host that you put into temporary maintenance mode is different from a host that you designate as a maintenance host. A maintenance host is a host that you dedicate for virtual machine maintenance tasks, such as the following:
  - Patching stored virtual machines and templates.
  - Staging scripted virtual machine creation before you move the virtual machines into your production environment.

## Examples

### 1: Place the specified host into maintenance mode and save all running virtual machines.

The first command gets the host object named VMHost01 and stores the object in the \$VMHost variable.

The second command places all running virtual machines that are deployed on the host stored in \$VMHost into a saved state. Then it sets the host status to "In Maintenance Mode".

```
PS C:\> $VMHost = Get-SCVMHost -ComputerName "VMHost01"
```

```
PS C:\> Disable-SCVMHost -VMHost $VMHost
```

### 2: Use live migration to migrate all running highly available virtual machines on a cluster node that is in maintenance mode.

The first command gets the host object named VMHost02 and stores the object in the \$VMHost variable.

The second command uses live migration to migrate all running highly available virtual machines on the host stored in \$VMHost to another node in the cluster. It places other running virtual machines into a saved state and then sets the value for the host state property to "In Maintenance Mode".

```
PS C:\> $VMHost = Get-SCVMHost -ComputerName "VMHost02"
```

```
PS C:\> Disable-SCVMHost -VMHost $VMHost -MoveWithinCluster
```

## Related topics

[Enable-SCVMHost](#)

[Move-SCVirtualMachine](#)

---

# Enable-SCLoadBalancerVIPMember

---

## Enable-SCLoadBalancerVIPMember

Enables a member of a load balancer VIP.

### Syntax

Parameter Set: SingleInstance

```
Enable-SCLoadBalancerVIPMember [-LoadBalancerVIPMember] <LoadBalancerVIPMember> [-  
JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer  
<ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Enable-SCLoadBalancerVIPMember enables a member of a load balancer virtual IP (VIP) and places the member in a Desired Status of Up.

To disable a VIP member, use the Disable-SCLoadBalancerVIPMember cmdlet.

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

#### **-LoadBalancerVIPMember<LoadBalancerVIPMember>**

Specifies a member of a virtual IP (VIP) in a load balancer.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

---

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **LoadBalancerVIPMember**

## Examples

### 1: Enable a member of a load balancer virtual IP (VIP).

The first command gets the load balancer object with the address of LB01.Contoso.com and stores the object in the \$LoadBalancer variable.

The second command gets the load balancer VIP with the IP address 192.168.0.1 for the load balancer stored in \$LoadBalancer and stores the object in the \$VIP variable.

The third command gets the VIP member for the load balancer VIP stored in \$VIP with the address of 192.168.0.1 and stores the object in the \$VIPMember variable.

The last command enables the VIP member stored in \$VIPMember and displays information about the VIP member to the user.

```
PS C:\> $LoadBalancer = Get-SCLoadBalancer -LoadBalancerAddress "LB01.Contoso.com"
PS C:\> $VIP = Get-SCLoadBalancerVIP -LoadBalancer $LoadBalancer -IPAddress "192.168.0.1"
PS C:\> $VIPMember = Get-SCLoadBalancerVIPMember -LoadBalancerVIP $VIP -IPAddress
"192.168.0.1"
PS C:\> Enable-SCLoadBalancerVIPMember -LoadBalancerVIPMember $VIPMember
```

## Related topics

[Disable-SCLoadBalancerVIPMember](#)

[Get-SCLoadBalancerVIP](#)



---

[Get-SCLoadBalancerVIPMember](#)

[New-SCLoadBalancerVIPMember](#)

[Remove-SCLoadBalancerVIPMember](#)

---

# Enable-SCRunAsAccount

---

## Enable-SCRunAsAccount

Enables a previously disabled Run As account.

### Syntax

Parameter Set: RunAsAccount

```
Enable-SCRunAsAccount [-RunAsAccount] <RunAsAccount> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: WithJobGroup

```
Enable-SCRunAsAccount -JobGroup <Guid> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Enable-SCRunAsAccount enables a Run As account in Virtual Machine Manager (VMM) that has been previously disabled using the Disable-SCRunAsAccount cmdlet.

### Parameters

#### **-JobGroup<Guid>**

Specifies an identifier for a series of commands that will run as a set just before the final command that includes the same job group identifier runs.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-RunAsAccount<RunAsAccount>**

Specifies a Run As account that contains credentials with permission to perform this action.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

---

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **RunAsAccount**

---

## Examples

### 1: Enable a Run As account.

The first command gets the Run As account object named RunAsAccount01 and stores the object in the \$RunAsAccount variable.

The second command enables the Run As account stored in \$RunAsAccount.

```
PS C:\> $RunAsAccount = Get-SCRunAsAccount -Name "RunAsAccount01"
```

```
PS C:\> Enable-SCRunAsAccount -RunAsAccount $RunAsAccount
```

## Related topics

[Get-SCRunAsAccount](#)

[Disable-SCRunAsAccount](#)

---

# Enable-SCVMHost

---

## Enable-SCVMHost

Restores a virtual machine host in maintenance mode to full service.

### Syntax

Parameter Set: Default

```
Enable-SCVMHost [-VMHost] <Host> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

### Detailed Description

The Enable-SCVMHost cmdlet restores a virtual machine host in maintenance mode to full service as a host managed by Virtual Machine Manager (VMM).

Enable-SCVMHost supports any type of host managed by VMM, including Hyper-V hosts, VMware ESX hosts, and Citrix XenServer hosts.

When you use Enable-SCVMHost to restore a host to full service, VMM automatically restores some capabilities but requires you to perform certain actions manually.

VMM automatically re-enables the following items:

- Creation of virtual machines on the host.
- Migration of virtual machines to the host.
- Host ratings for the host.

VMM also automatically refreshes the host to its current state.

You must manually restart any virtual machines that are in a saved state on a standalone host. For hosts that are a node in a host cluster, you must manually restart the virtual machines and manually move any migrated virtual machines back to this node that you migrated to another node when you placed the host into maintenance mode.

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMHost<Host>**

Specifies a virtual machine host object. VMM supports Hyper-V hosts, VMware ESX hosts, and Citrix XenServer hosts.

For more information about each type of host, type: "Get-Help Add-SCVMHost -detailed". See the examples for a specific cmdlet to determine how that cmdlet uses this parameter.

---

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **Host**
- **Host[]**

## Notes

- A host that you put into temporary maintenance mode is different from a host that you designate as a maintenance host. A maintenance host is a host that you dedicate for virtual machine maintenance tasks, such as the following tasks:
  - Patching stored virtual machines and templates.
  - Staging scripted virtual machine creation before you move the virtual machines into your production environment.

## Examples

### 1: Restores the specified host to service.

The first command gets host object named VMHost01 and stores the object in the \$VMHost variable.

The second unblocks virtual machine creation operations on the host stored in \$VMHost, and includes the host in host ratings during placement. It also takes the host status out of "In Maintenance Mode" so that the next host refresh job will set the host status to its current state.

```
PS C:\> $VMHost = Get-SCVMHost -ComputerName "VMHost01"
```

```
PS C:\> Enable-SCVMHost -VMHost $VMHost
```



---

## Related topics

[Disable-SCVMHost](#)

[Move-SCVirtualMachine](#)

---

# Expand-SCVirtualDiskDrive

---

## Expand-SCVirtualDiskDrive

Expands a virtual hard disk attached to a virtual disk drive object.

### Syntax

Parameter Set: Default

```
Expand-SCVirtualDiskDrive [-VirtualDiskDrive] <VirtualDiskDrive> -VirtualHardDiskSizeGB  
<Int32> [-JobGroup <Guid> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-  
RunAsynchronously] [ <CommonParameters>]
```

### Detailed Description

The Expand-SCVirtualDiskDrive cmdlet expands a virtual hard disk attached to a virtual disk drive object in order to increase the total capacity of the virtual hard disk. The virtual machine must be deployed on a host managed by Virtual Machine Manager (VMM) and must be in a stopped state before you can expand the virtual hard disk.

You can use the Expand-SCVirtualDiskDrive cmdlet to expand a Windows-based virtual hard disk file (a .vhd or .vhdx file) attached to a virtual disk drive object on a virtual machine deployed on a Hyper-V host. You can also use this cmdlet to expand a VMware-based virtual hard disk file (a .vmdk file) on a virtual machine deployed on an ESX host. However, you cannot use this cmdlet to expand a virtual hard disk on a virtual machine deployed on a Citrix XenServer host.

### Parameters

#### **-JobGroup<Guid>**

Specifies an identifier for a series of commands that will run as a set just before the final command that includes the same job group identifier runs.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false

---

Accept Wildcard Characters?	false
-----------------------------	-------

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false

Accept Wildcard Characters?	false
-----------------------------	-------

## **-VirtualDiskDrive<VirtualDiskDrive>**

Specifies a virtual disk drive object. You can attach either a virtual hard disk (for a virtual machine on any host) or a pass-through disk (for a virtual machine on a Hyper-V host or an ESX host) to a virtual disk drive object.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **-VirtualHardDiskSizeGB<Int32>**

Specifies, in gigabytes (GB), the size to which a dynamically expanding virtual hard disk will expand.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **VirtualHardDisk**

---

## Notes

- Requires a VMM virtual disk drive object, which can be retrieved by using the Get-SCVirtualDiskDrive cmdlet.

## Examples

### 1. Expand a virtual hard disk attached to a virtual disk drive on a virtual machine.

The first command gets the virtual machine object named VM01 and stores the object in the \$VM variable.

The second command gets the virtual disk drive object located on the first controller (\$\_.Bus -eq 0) and first slot of that controller (\$\_.Lun -eq 0) of VM01, and stores the object in the \$VirtDiskDrive variable.

The last command expands the size of the virtual hard disk attached to the virtual disk drive to 40 GB.

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM01"
```

```
PS C:\> $VirtDiskDrive = Get-SCVirtualDiskDrive -VM $VM | where {$_.Bus -eq 0 -and $_.Lun -eq 0}
```

```
PS C:\> Expand-SCVirtualDiskDrive -VirtualDiskDrive $VirtDiskDrive -VirtualHardDiskSizeGB 40
```

## Related topics

[Compress-SCVirtualDiskDrive](#)

[Convert-SCVirtualDiskDrive](#)

[Get-SCVirtualDiskDrive](#)

[New-SCVirtualDiskDrive](#)

[Remove-SCVirtualDiskDrive](#)

[Set-SCVirtualDiskDrive](#)

---

# Export-SCLibraryPhysicalResource

---

## Export-SCLibraryPhysicalResource

Exports a library resource form a VMM library share to a local file folder or network share.

### Syntax

Parameter Set: Default

```
Export-SCLibraryPhysicalResource [-Path] <String> -Resource <ItemBase> [-  
AllowUnencryptedTransfer] [-OverwriteExistingFiles] [-VMMServer <ServerConnection> ] [  
<CommonParameters>]
```

### Detailed Description

The Export-SCLibraryPhysicalResource exports (copies) a library resource from a Virtual Machine Manager (VMM) library share to a local file folder or network share.

### Parameters

#### -AllowUnencryptedTransfer

Indicates that network file transfers do not require encryption. Allowing unencrypted network file transfers can improve performance if neither the source host nor the destination host requires encryption.

Use this parameter to:

- Allow unencrypted file transfers into, or out of, the library.
- Allow unencrypted file transfers into, out of, or within a host group.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-OverwriteExistingFiles**

Indicates that files with the same name are overwritten when importing or exporting resources into or out of the VMM library.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Path<String>**

Specifies the destination path for the operation.

Example formats:

Local path -Path "F:\"

UNC path -Path "\\Library\Templates"

Volume GUID path -Path "\\?\Volume{4703c1ea-8ae7-11db-b473-00123f7603e3}\"

VMware ESX path -Path "[storage1]\MyVMwareFolderForVMs\MyVM.vmx"

Citrix XenServer path - Path "Local storage[99b6212f-b63d-c676-25f9-d6c460992de7]"

Wildcards are supported for "Get" cmdlets and when you specify the UNC path:

Example format:

UNC path -Path "\\VMHostServer\MyVMs\\*VM\*"

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Resource<ItemBase>**

Specifies a resource object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Examples**

### **1. Export a specific library resource to a local file folder.**

The first command gets the virtual hard disk object named VHD01 and stores the object in the \$Resource variable.

The second command exports the resource stored in \$Resource to the folder named C:\ExportedLibResources. If the resource had been previously exported to this folder, this command would overwrite any of those files.

```
PS C:\> $Resource = Get-SCVirtualHardDisk -Name "VHD01.vhd"
```

```
PS C:\> Export-SCLibraryPhysicalResource -Resource $Resource -Path "C:\ExportedLibResources"
-OverwriteExistingFiles
```



---

## 2. Export a library resource to a network share.

The first command gets the application package object named WebApp01.zip and stores the object in the \$Resource variable.

The second command exports the library resource stored in \$Resource to the network share named \\FileShare01\LibExports.

```
PS C:\> $Resource = Get-SCApplicationPackage -Name "WebApp01.zip"
```

```
PS C:\> Export-SCLibraryPhysicalResource -Resource $Resource -Path  
"\\FileShare01\LibExports"
```

### Related topics

[Import-SCLibraryPhysicalResource](#)

---

# Export-SCTemplate

---

## Export-SCTemplate

Exports a template from the VMM library to the specified path.

### Syntax

Parameter Set: ServiceTemplate

```
Export-SCTemplate [-ServiceTemplate] <ServiceTemplate> -Path <String> [-AllowUnencryptedTransfer] [-IncludeAllLibraryResources] [-IncludeLibraryResources <ItemBase[]> ] [-Overwrite] [-Password <String> ] [-SettingsIncludePrivate] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: VMTemplate

```
Export-SCTemplate [-VMTemplate] <Template> -Path <String> [-AllowUnencryptedTransfer] [-IncludeAllLibraryResources] [-IncludeLibraryResources <ItemBase[]> ] [-Overwrite] [-Password <String> ] [-SettingsIncludePrivate] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Export-SCTemplate cmdlet exports a template from the VMM library to the specified path. You can also export the library objects on which the template is dependent.

### Parameters

#### -AllowUnencryptedTransfer

Indicates that network file transfers do not require encryption. Allowing unencrypted network file transfers can improve performance if neither the source host nor the destination host requires encryption.

Use this parameter to:

- Allow unencrypted file transfers into, or out of, the library.
- Allow unencrypted file transfers into, out of, or within a host group.

Aliases	none
Required?	false
Position?	named

Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-IncludeAllLibraryResources**

Indicates that all of the dependencies for a template are exported from the VMM library with the template.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-IncludeLibraryResources<ItemBase[]>**

Specifies dependent library resources that are to be exported with a template.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Overwrite**

Indicates that an import or export operation will overwrite an existing file with the same name. Or, that an import operation will overwrite an existing virtual machine template or service template object with the same name.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Password<String>**

Specifies a secure string that contains a password.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Path<String>**

Specifies the destination path for the operation.

Example formats:

Local path -Path "F:\"

UNC path -Path "\\Library\Templates"

Volume GUID path -Path "\\?\Volume{4703c1ea-8ae7-11db-b473-00123f7603e3}\"

VMware ESX path -Path "[storage1]\MyVMwareFolderForVMs\MyVM.vmx"

Citrix XenServer path - Path "Local storage[99b6212f-b63d-c676-25f9-d6c460992de7]"

Wildcards are supported for "Get" cmdlets and when you specify the UNC path:

Example format:

UNC path -Path "\\VMHostServer\MyVMs\\*VM\*"

Aliases	none
Required?	true
Position?	named
Default Value	none

Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-ServiceTemplate<ServiceTemplate>**

Specifies a service template object.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-SettingsIncludePrivate**

Indicates that sensitive template settings are included in an import or export operation.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **-VMTemplate<Template>**

Specifies a VMM template object used to create virtual machines.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **SCTemplate**

## **Examples**

### **1: Export a service template with all of its settings.**

The first command gets the service template object named ServiceTemplate01 and stores the object in the \$ServiceTemplate variable.

The second command exports the service template stored in \$ServiceTemplate, including all settings, and overwrites existing template export packages that have the same name.

```
PS C:\> $ServiceTemplate = Get-SCServiceTemplate -Name "ServiceTemplate01"
```

```
PS C:\> Export-SCTemplate -ServiceTemplate $ServiceTemplate -Path "C:\TemplateExports" -SettingsIncludePrivate -Overwrite
```

---

## 2: Export multiple service templates with all of their settings.

This command uses the `Get-SCServiceTemplate` cmdlet to get all service template objects. Then, it uses the pipeline operator to send the objects to the `Export-SCTemplate` cmdlet which exports the templates, overwriting any existing files.

```
PS C:\> Get-SCServiceTemplate | Export-SCTemplate -Path "C:\TemplateExports" -  
SettingsIncludePrivate -Overwrite
```

## 3. Export a service template including its dependent library resources.

The first command gets the service template object named `ServiceTemplate01` and stores the object in the `$ServiceTemplate` variable.

The second command exports `ServiceTemplate01` and all of its dependent resources from the VMM library to `C:\TempalteExports`.

```
PS C:\> $ServiceTemplate = Get-SCServiceTemplate -Name "ServiceTemplate01"  
PS C:\> Export-SCTemplate -ServiceTemplate $ServiceTemplate -Path "C:\TemplateExports" -  
IncludeAllLibraryResources
```

## Related topics

[Import-SCTemplate](#)

---

# Find-SCCluster

---

## Find-SCCluster

Finds the specified failover cluster in a VMM environment.

### Syntax

Parameter Set: UserCredentials

```
Find-SCCluster [-ComputerName] <String> -Credential <VMMServerCredential> [-EnumerateFileServers <Boolean> ] [-JobVariable <String> ] [-NonTrustedDomainHost] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ServerCredentials

```
Find-SCCluster -LibraryServer <LibraryServer> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: XenServerHost

```
Find-SCCluster [-ComputerName] <String> -Credential <VMMServerCredential> -XenServerHost [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-TCPPort <UInt32> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Find-SCCluster cmdlet queries Virtual Machine Manager (VMM) for the specified failover cluster or one of its nodes. If the cluster is found, VMM returns an object that contains more information about the failover cluster. Information returned by Find-SCCluster includes cluster name, nodes of the cluster, and highly available file servers hosted by the cluster.

### Parameters

#### **-ComputerName<String>**

Specifies the name of a computer that VMM can uniquely identify on your network. Valid formats are: FQDN, IPv4 or IPv6 address, or NetBIOS name.

NOTE: See the examples for a specific cmdlet to determine how that cmdlet specifies the computer name.

Aliases	none
Required?	true



Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Credential<VMMCredential>**

Specifies a credential object or, for some cmdlets, a Run As account object that contains the user name and password of an account that has permission to perform this action. Or, in the case of Restart-SCJob, has permission to complete a restarted task.

For more information about the PSCredential object, type: "Get-Help Get-Credential".

For more information about Run As accounts, type: "Get-Help New-SCRunAsAccount".

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-EnumerateFileServers<Boolean>**

Indicates whether the file servers are listed.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

---

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-LibraryServer<LibraryServer>**

Specifies a VMM library server object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-NonTrustedDomainHost**

Indicates that the host to be added to VMM belongs to a non-trusted domain.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

---

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-TCPPort<UInt32>**

Specifies a numeric value that represents a TCP port.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **-XenServerHost**

Indicates that the specified host is a Citrix XenServer host.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **Cluster**

## **Examples**

### **1: Find all nodes of a failover cluster from the cluster name.**

The first command gets the Run As account object named RunAsAccount01 and stores the object in the \$Credential variable.

---

The second command queries VMM for the failover cluster named VMHostCluster01 and stores the cluster object in \$Cluster, using \$Credential to provide the Run As account to Find-SCCluster. The ComputerName parameter treats the name of the cluster as if it were the name of a computer.

The last command displays the FQDNs of the cluster nodes to the user.

```
PS C:\> $Credential = Get-SCRunAsAccount -Name "RunAsAccount01"
```

```
PS C:\> $Cluster = Find-SCCluster -ComputerName "VMHostCluster01.Contoso.com" -Credential $Credential
```

```
PS C:\> $Cluster.ClusterNodes
```

## 2: Find all nodes of a failover cluster from one of the node names.

The first command gets the Run As account object named RunAsAccount01 and stores the object in the \$Credential variable.

The second command queries VMM for a failover cluster node named VMHostNode02 and stores the returned cluster object in \$Cluster.

The third command displays the FQDN of the cluster to the user.

The last command displays the FQDN of each node in the cluster to the user.

```
PS C:\> $Credential = Get-SCRunAsAccount -Name "RunAsAccount01"
```

```
PS C:\> $Cluster = Find-SCCluster -ComputerName "VMHostNode02.Contoso.com" -Credential $Credential
```

```
PS C:\> $Cluster.Name
```

```
PS C:\> $Cluster.ClusterNodes
```

## 3: Find, by using the cluster name, all highly available file servers hosted by that failover cluster.

The first command gets the RunAs account object RunAsAccount01 and stores the object in the \$Credential variable.

The second command queries VMM for the failover cluster named VMHostCluster03 and stores the cluster object in \$Cluster.

The third command displays the FQDNs of all highly available file servers hosted by the failover cluster stored in \$Cluster. This example assumes that the failover cluster is hosting at least one highly available file server.

```
PS C:\> $Credential = Get-SCRunAsAccount -Name "RunAsAccount01"
```

```
PS C:\> $Cluster = Find-SCCluster -ComputerName "VMHostCluster03.Contoso.com" -Credential $Credential
```

```
PS C:\> $Cluster.HAFileServers
```

## Related topics

[Add-SCVMHostCluster](#)

---

[Get-SCVMHostCluster](#)

[Install-SCVMHostCluster](#)

[Move-SCVMHostCluster](#)

[Read-SCVMHostCluster](#)

[Remove-SCVMHostCluster](#)

[Set-SCVMHostCluster](#)

[Test-SCVMHostCluster](#)

[Uninstall-SCVMHostCluster](#)

---

# Find-SCComputer

---

## Find-SCComputer

Searches for computers by querying Active Directory, and returns the computer objects.

### Syntax

Parameter Set: SearchComputersByNameFilter

```
Find-SCComputer -Credential <VMCrediential> -Domain <String> [-ComputerNameFilter <String> ]  
[-DiscoveryID <Guid> ] [-ExcludeVMMHost] [-ExcludeVMMLibrary] [-ExcludeVMS] [-  
FindHyperVHost] [-JobVariable <String> ] [-RunAsynchronously] [-VMMServer <ServerConnection>  
] [ <CommonParameters>]
```

Parameter Set: SearchBMByIPAddressRange

```
Find-SCComputer -BMCProtocol {None | IPMI | SMASH | Custom} -BMCRunAsAccount <RunAsAccount>  
-IPAddressRangeEnd <String> -IPAddressRangeStart <String> [-All] [-  
BMCCustomConfigurationProvider <ConfigurationProvider> ] [-BMCPort <Int32> ] [-DiscoveryID  
<Guid> ] [-JobVariable <String> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [  
<CommonParameters>]
```

Parameter Set: SearchBMByIPSubnet

```
Find-SCComputer -BMCProtocol {None | IPMI | SMASH | Custom} -BMCRunAsAccount <RunAsAccount>  
-Subnet <String> [-All] [-BMCCustomConfigurationProvider <ConfigurationProvider> ] [-BMCPort  
<Int32> ] [-DiscoveryID <Guid> ] [-JobVariable <String> ] [-RunAsynchronously] [-VMMServer  
<ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: SearchComputersByADQuery

```
Find-SCComputer -ADSearchFilter <String> -Credential <VMCrediential> -Domain <String> [-  
DiscoveryID <Guid> ] [-ExcludeVMMHost] [-ExcludeVMMLibrary] [-ExcludeVMS] [-FindHyperVHost]  
[-JobVariable <String> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [  
<CommonParameters>]
```

Parameter Set: ValidateBMC

```
Find-SCComputer -BMCAddress <String> -BMCProtocol {None | IPMI | SMASH | Custom} -  
BMCRunAsAccount <RunAsAccount> [-BMCCustomConfigurationProvider <ConfigurationProvider> ] [-  
BMCPort <Int32> ] [-DeepDiscovery] [-DiscoveryID <Guid> ] [-JobVariable <String> ] [-  
RunAsynchronously] [-SMBiosGuid <Guid> ] [-VMMServer <ServerConnection> ] [  
<CommonParameters>]
```

Parameter Set: ValidateComputer

```
Find-SCComputer [-ComputerName] <String> [-Credential <VMCrediential> ] [-DiscoveryID <Guid>  
] [-ExcludeVMMHost] [-ExcludeVMMLibrary] [-RunAsynchronously] [-VMMServer <ServerConnection>  
] [ <CommonParameters>]
```

---

## Detailed Description

Searches for one or more computers by querying Active Directory, and returns the computer objects. You can use this cmdlet to query Active Directory for computers based on specified criteria, or a combination of criteria, including:

- The fully qualified domain name (FQDN) of a computer
- All or part of the computer name
- The name of a domain
- All computers except hosts managed by Virtual Machine Manager (VMM)
- All computers except library servers managed by VMM
- Only Hyper-V hosts
- Bare-metal computers that have out-of-band controllers

NOTE: If you add a new computer (such as a host or library server) located in an Active Directory domain to VMM and then immediately run the Discover-Computer cmdlet, the cmdlet might not immediately discover the new computer when it searches Active Directory. This delay may occur because data about the new computer might not have replicated yet across the Active Directory domain. If you are a Domain Administrator, you can use the Active Directory Sites and Services console to force the data to replicate immediately.

## Parameters

### **-ADSearchFilter<String>**

Defines an Active Directory query for discovery. Use this parameter to specify a query that contains Active Directory domain information and search filters.

Example format: -ADSearchFilter

```
"(&(sAMAccountType=805306369)(name=katarina*)(objectCategory=computer)(objectClass=computer)(operatingSystem=Windows\20Server*))"
```

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false



---

## **-All**

Retrieves a full list of all subordinate objects independent of the parent object. For example, the command `Get-SCVirtualDiskDrive -All` retrieves all virtual disk drive objects regardless of the virtual machine object or template object that each virtual disk drive object is associated with.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-BMCAddress<String>**

Specifies, or updates, the out-of-band baseboard management controller (BMC) address for a specific physical machine. This might be an IP address, the fully qualified domain name (FQDN), or the DNS prefix (which is usually the same name as the NetBIOS name).

Typically, the BMC address and its connection to the network are separate from the IP address associated with a standard network adapter. Alternatively, some computers do use a standard network adapter to provide a single address for the BMC and for the network adapter. However, the BMC address has a unique port and is thus uniquely identifiable on the network.

Example IPv4 format: `-BMCAddress "10.0.0.21"`

Example Ipv6 format: `-BMCAddress "2001:4898:2a:3:657b:9c7a:e1f0:6829"`

Example FQDN format: `-BMCAddress "Computer01.Contoso.com"`

Example NetBIOS format: `-BMCAddress "Computer01"`

NOTE: By default, VMM uses an IP address or FQDN for the BMCAddress. However, it is also possible to create a Windows PowerShell module that enables you to specify other types of addresses as the BMC address.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-BMCCustomConfigurationProvider<ConfigurationProvider>**

Specifies, or updates, a configuration provider object for a baseboard management controller (BMC). A configuration provider is a plug-in to VMM that translates VMM PowerShell commands to API calls that are specific to a type of baseboard management controller. This parameter should be used with the Custom BMCProtocol.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-BMCPort<Int32>**

Specifies, or updates, the out-of-band baseboard management controller (BMC) port for a specific physical machine. A BMC port is also known as a service processor port. Example default ports are 623 for IPMI and 443 for SMASH over WS-Man.

Example format: -BMCPort 80

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-BMCProtocol<OutOfBandManagementType>**

Specifies, or updates, the protocol that VMM uses to communicate with the out-of-band baseboard management controller (BMC). Valid values are: IPMI, SMASH, Custom.

A BMC (also known as a service processor or management controller) is a specialized controller on the motherboard of a server that acts an interface between the hardware and system management software. If the motherboard of a physical machine includes a BMC, when the machine is plugged in (whether it is powered off or powered on, and whether or not an operating system is installed), information about system hardware and the state of that system hardware health is available.

Example format: -BMCProtocol "Custom"

---

NOTE: The Custom protocol requires using the BMCCustomConfigurationProvider.

The acceptable values for this parameter are:

None	
IPMI	
SMASH	
Custom	

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-BMCRunAsAccount<RunAsAccount>**

Specifies the Run As account to use with the baseboard management controller (BMC) device.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-ComputerName<String>**

Specifies the name of a computer that VMM can uniquely identify on your network. Valid formats are: FQDN, IPv4 or IPv6 address, or NetBIOS name.

NOTE: See the examples for a specific cmdlet to determine how that cmdlet specifies the computer name.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-ComputerNameFilter<String>**

Specifies the partial or full name of a computer that the cmdlet will try to discover in Active Directory.

Example format: -ComputerNameFilter "host"

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Credential<VMMSCredential>**

Specifies a credential object or, for some cmdlets, a Run As account object that contains the user name and password of an account that has permission to perform this action. Or, in the case of Restart-SCJob, has permission to complete a restarted task.

For more information about the PSCredential object, type: "Get-Help Get-Credential".

For more information about Run As accounts, type: "Get-Help New-SCRunAsAccount".

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-DeepDiscovery**

Indicates that deep discovery is used for the specified computer.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-DiscoveryID<Guid>**

For internal use only (not for use in your code).

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Domain<String>**

Specifies a fully qualified domain name (FQDN) for an Active Directory domain.

Example format: -Domain "Domain01.Corp.Contoso.com"

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-ExcludeVMMHost**

Excludes virtual machine hosts currently managed by VMM.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-ExcludeVMMLibrary**

Excludes library servers currently managed by VMM.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-ExcludeVMs**

Excludes virtual machines.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-FindHyperVHost**

Searches for computers running Windows Server 2008 or later on which the Hyper-V server role is enabled.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-IPAddressRangeEnd<String>**

Specifies the last IP address in a range of IP addresses. Use with IPAddressRangeStart.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-IPAddressRangeStart<String>**

Specifies the first IP address in a range of IP addresses. Use with IPAddressRangeEnd.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-SMBiosGuid<Guid>**

Specifies the System Management BIOS globally unique identifier (SMBIOS GUID) for a physical computer that is associated with a record for that physical computer in VMM. SMBIOS defines data structures and access methods that enable a user or application to store and retrieve information about hardware on this computer, such as the name of the system, manufacturer, or the system BIOS version. Windows operating systems retrieve SMBIOS data at system startup and make that data available to programs.

Aliases	none
Required?	false
Position?	named
Default Value	none



Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Subnet<String>**

Specifies an IP subnet (IPv4 or IPv6) in Classless Inter-Domain Routing (CIDR) notation.

Example format for an IPv4 subnet: 192.168.0.1/24

Example format for an IPv6 subnet: FD4A:29CD:184F:3A2C::/64

NOTE: An IP subnet cannot overlap with any other subnet in a host group or child host groups.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

---

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **Computer**

## Examples

### 1: Search for computers in a specific domain that meet the specified criteria.

The first command gets the Run As account object named Host Computer Account 01 and stores the object in the \$RunAsAccount variable.

The second command queries Active Directory and returns a list of Hyper-V computer objects for computers that are located in the Contoso.com domain, that have a name starting with "host", and that are not managed by VMM. As this command is processed, \$RunAsAccount provides credentials to Find-SCComputer.

NOTE: When you use Find-SCComputer with the Domain parameter, you must specify the fully qualified domain name.

```
PS C:\> $RunAsAccount = Get-SCRunAsAccount -Name "Host Computer Account 01"
```

```
PS C:\> Find-SCComputer -ComputerNameFilter "host" -Domain "Contoso.com" -FindHyperVHosts -  
ExcludeVMMHost -RunAsAccount $RunAsAccount
```

### 2: Search for a specific computer by name and validate its properties in Active Directory.

This command uses the fully qualified domain name (FQDN) of the computer named VMHost01 to find this computer in Active Directory, returns the computer object, and displays the computer object properties to the user.

```
PS C:\> Find-SCComputer -ComputerName "VMHost01.Contoso.com"
```

### 3: Search for all computers in the specified domain that are not a VMM library server.

The first command gets the Run As account object named Host Computer Account 01 and stores the object in the \$RunAsAccount variable.

The second command queries Active Directory for all computers in the Contoso.com domain that include "vmm" in the computer name except for VMM library servers. As this command is processed, \$RunAsAccount provides credentials to Find-SCComputer.

```
PS C:\> $RunAsAccount = Get-SCRunAsAccount -Name "Host Computer Account 01"
```

```
PS C:\> Find-SCComputer -ComputerNameFilter "vmm" -Domain "Contoso.com" -ExcludeVMMLibrary -  
RunAsAccount $RunAsAccount
```

---

#### 4: Find all unmanaged computers in the specified domain by using an Active Directory query.

The first command gets the Run As account object named Host Computer Account 01 and stores the object in the \$RunAsAccount variable.

The second command queries Active Directory for all computers in the Contoso.com domain that meet the query criteria: any Windows Server 2008 R2 computer with a name that starts with Test0 but which are not VMM library servers or VMM hosts. As this command is processed, \$RunAsAccount provides credentials to Find-SCComputer.

```
PS C:\> $RunAsAccount = Get-SCRunAsAccount -Name "Host Computer Account 01"
PS C:\> Find-SCComputer -ADSearchFilter
"(&(sAMAccountType=805306369)(name=Test0*)(objectCategory=computer)(objectClass=computer)(operatingSystem=Windows\20Server\202008\20R2*))" -Domain "Contoso.com" -ExcludeVMMLibrary -
ExcludeVMMHost -RunAsAccount $RunAsAccount
```

#### 5: Find bare-metal computers that have out-of-band controllers that are within a specific network range.

The first command gets the Run As account object named BMC Account 01 and stores the object in the \$BMCRunAsAccount variable.

The second command scans the network for a physical computer with the specified BMCAddress. As this command is processed, \$BMCRunAsAccount provides credentials to Find-SCComputer.

```
PS C:\> $BMCRunAsAccount = Get-SCRunAsAccount "BMC Account 01"
PS C:\> Find-SCComputer -BMCAddress "10.10.0.1" -BMCRunAsAccount $BMCRunAsAccount -
BMCProtocol "IPMI"
```

#### 6: Perform deep discovery on a physical machine with BMC support.

The first command gets the Run As account object named BMC Account 01 and stores the object in the \$BMCRunAsAccount variable.

The second command scans the network for a physical computer with the specified BMCAddress. As this command is processed, \$BMCRunAsAccount provides credentials to Find-SCComputer.

The third command performs the deep discovery on the computer object and returns the physical machine object to \$MyComputer01 variable.

The last command shows the properties of the computer object stored in the \$MyComputer01 variable.

```
PS C:\> # Discover bare metal computers from an IP address "10.10.0.1"
PS C:\> $BMCRunAsAccount = Get-SCRunAsAccount "BMC Account 01"
PS C:\> $MyComputer01 = Find-SCComputer -BMCAddress "10.10.0.1" -BMCRunAsAccount
$BMCRunAsAccount -BMCProtocol "IPMI"
# Perform Deep Discovery operation on this computer
PS C:\> $MyComputer01 = Find-SCComputer -DeepDiscovery -BMCAddress $MyComputer01.BMCAddress
-BMCRunAsAccount $BMCRunAsAccount -BMCProtocol "IPMI" -SMBIOSGUID $MyComputer01.SMBIOSGUID
# Exam the computer properties and we should see NIC and disk info now.
```

---

PS C:\> \$MyComputer01

## Related topics

[Add-SCVMHost](#)

[New-SCVMHost](#)

---

# Find-SCJob

---

## Find-SCJob

Searches for VMM jobs.

### Syntax

Parameter Set: FindAfter

```
Find-SCJob -MaxCount <Int32> [-After <DateTime> ] [-ObjectID <Guid> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: FindRecent

```
Find-SCJob -MaxCount <Int32> -Recent [-ObjectID <Guid> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Find-SCJob cmdlet searches for Virtual Machine Manager (VMM) jobs. You can search for the most recent jobs, jobs after a certain date, or search for a job by its result or target ID.

### Parameters

#### **-After<DateTime>**

Specifies a DateTime object for the time after which jobs are retrieved. By default, time is represented in the local time zone. You can pass the DateTime object as UTC for standardized time.

When this parameter is specified, jobs are filtered by the StartTime property and returned from earliest start time to latest start time.

For information about getting a DateTime object, see the Get-Date cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-MaxCount<Int32>**

Specifies the maximum number of jobs to return.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-ObjectID<Guid>**

Specifies the object ID, as a GUID, for a service. This ID is the result object or target object for the returned job.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Recent**

Indicates that the most recent jobs are returned. Jobs are returned from latest start time to earliest start time.

Use this parameter with MaxCount to retrieve only a specified number of jobs.

Aliases	none
Required?	true
Position?	named
Default Value	none

Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- Job

## **Examples**

### **1: Find all jobs for the last five hours.**

This command searches for all jobs that started within the last five hours. Because the MaxCount parameter value is set to 25, only 25 job objects are displayed.

```
PS C:\> $StartTime = (Get-Date) - (New-TimeSpan -Hours 5)
```

```
PS C:\> Find-SCJob -MaxCount 25 -After $StartTime
```

### **2: Find recent jobs.**

This command searches for recent jobs and displays the 50 most recent jobs.

```
PS C:\> Find-SCJob -MaxCount 50 -Recent
```

---

### 3: Find jobs by ObjectID.

The first command gets the virtual machine object named VM01 and stores the object in the \$VM variable.

The second command returns up to 10 jobs for the specified virtual machine ID.

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM01"
```

```
PS C:\> Find-SCJob -MaxCount 10 -ObjectID $VM.Id
```

### Related topics

[Get-SCJob](#)

[Restart-SCJob](#)

[Stop-SCJob](#)



---

# Find-SCLibraryShare

---

## Find-SCLibraryShare

Finds all of the shares on the specified computer or library server that can be added as library shares.

### Syntax

Parameter Set: New

```
Find-SCLibraryShare [-ComputerName] <String> -Credential <VMMCredential> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: Existing

```
Find-SCLibraryShare -LibraryServer <LibraryServer> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Find-SCLibraryShare cmdlet finds all of the shares on the specified computer or library server that can be added as library shares. For library servers, this command also returns shares that are already Virtual Machine Manager (VMM) library shares.

### Parameters

#### **-ComputerName<String>**

Specifies the name of a computer that VMM can uniquely identify on your network. Valid formats are: FQDN, IPv4 or IPv6 address, or NetBIOS name.

NOTE: See the examples for a specific cmdlet to determine how that cmdlet specifies the computer name.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-Credential<VMMSCredential>**

Specifies a credential object or, for some cmdlets, a Run As account object that contains the user name and password of an account that has permission to perform this action. Or, in the case of Restart-SCJob, has permission to complete a restarted task.

For more information about the PSCredential object, type: "Get-Help Get-Credential".

For more information about Run As accounts, type: "Get-Help New-SCRunAsAccount".

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-LibraryServer<LibraryServer>**

Specifies a VMM library server object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named

---

Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **DiscoveredShare[]**

## Examples

### 1: Find Windows shares on a computer that is not yet a VMM library server.

The first command uses Get-Credential to prompt you to supply a user name and password with permissions to access Windows shares on Server01 and stores your credentials in the \$Credential variable.

The second command confirms that you have valid credentials for this operation and then displays all existing Windows shares capable of becoming VMM library shares.

```
PS C:\> $Credential = Get-Credential
```

```
PS C:\> Find-SCLibraryShare -Credential $Credential -ComputerName "Server01.Contoso.com"
```

### 2: Find shares on a VMM library server.

This command displays all Windows shares capable of becoming library shares that exist on LibraryServer01 as well as all shares that are already VMM library shares.

```
PS C:\> Find-SCLibraryShare -LibraryServer "LibraryServer01.Contoso.com"
```

## Related topics

[Add-SCLibraryServer](#)

[Add-SCLibraryShare](#)

---

# Get-SCApplicableVMHostGroup

---

## Get-SCApplicableVMHostGroup

Gets an applicable host group.

### Syntax

Parameter Set: ByLogicalSwitch

```
Get-SCApplicableVMHostGroup -LogicalSwitch <LogicalSwitch> [-VMMServer <ServerConnection> ]  
[ <CommonParameters>]
```

Parameter Set: ByNativeUplinkPortProfile

```
Get-SCApplicableVMHostGroup -NativeUplinkPortProfile <NativeUplinkPortProfile> [-VMMServer  
<ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ByUplinkPortProfileSet

```
Get-SCApplicableVMHostGroup -UplinkPortProfileSet <UplinkPortProfileSet> [-VMMServer  
<ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCApplicableVMHostGroup cmdlet gets the host group associated with a logical switch, a native uplink port profile, or an uplink port profile set

### Parameters

#### **-LogicalSwitch<LogicalSwitch>**

Specifies a logical switch object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-NativeUplinkPortProfile<NativeUplinkPortProfile>**

Specifies a native uplink port profile object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-UplinkPortProfileSet<UplinkPortProfileSet>**

Specifies an uplink port profile set object.

To obtain an uplink port profile set object, use the Get-SCUplinkPortProfileSet cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Examples

### 1: Get the applicable host group for a logical switch.

The first command gets the logical switch object named LogicalSwitch01 and stores the object in the \$LogSwitch variable.

The second command gets the applicable host group for the logical switch stored in \$LogSwitch.

```
PS C:\> $LogSwitch = Get-SCLogicalSwitch -Name "LogicalSwitch01"
```

```
PS C:\> Get-SCApplicableVMHostGroup -LogicalSwitch $LogSwitch
```

### 2: Get the applicable host group for an uplink port profile set.

The first command gets the uplink port profile set object named UplinkPortProfileSet01 and stores the object in the \$UplinkPortProfSet variable.

The second command gets the applicable host group for the uplink port profile set stored in \$UplinkPortProfSet.

```
PS C:\> $UplinkPortProfSet = Get-SCUplinkPortProfileSet -Name "UplinkPortProfileSet01"
```

```
PS C:\> Get-SCApplicableVMHostGroup -UplinkPortProfileSet $UplinkPortProfSet
```

### 3: Get the applicable host group for a NativeUplinkPortProfile.

The first command gets the native uplink port profile object named NativeUplinkPortProfile01 and stores the object in the \$NativeUplinkPortProf variable.

The second command gets the applicable host group for the native uplink port profile stored in \$NativeUplinkPortProf.

```
PS C:\> $NativeUplinkPortProf = Get-SCNativeUplinkPortProfile -Name  
"NativeUplinkPortProfile01"
```

```
PS C:\> Get-SCApplicableVMHostGroup -NativeUplinkPortProfile $NativeUplinkPortProf
```

## Related topics

[Get-SCLogicalSwitch](#)

[Get-SCNativeUplinkPortProfile](#)

[Get-SCUplinkPortProfileSet](#)

---

# Get-SCApplication

---

## Get-SCApplication

Gets the applications that are currently installed within a virtual machine that were installed by VMM.

### Syntax

Parameter Set: All

```
Get-SCApplication [-All] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ApplicationHost

```
Get-SCApplication -ApplicationHost <ApplicationHost> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ID

```
Get-SCApplication [-ID <Guid> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: VM

```
Get-SCApplication -VM <VM> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

## Detailed Description

The Get-SCApplication cmdlet gets the applications installed on a virtual machine by Virtual Machine Manager (VMM).

## Parameters

### -All

Retrieves a full list of all subordinate objects independent of the parent object. For example, the command `Get-SCVirtualDiskDrive -All` retrieves all virtual disk drive objects regardless of the virtual machine object or template object that each virtual disk drive object is associated with.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-ApplicationHost<ApplicationHost>**

Specifies an application host object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-ID<Guid>**

Specifies the numerical identifier (as a globally unique identifier, or GUID) for a specific object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VM<VM>**

Specifies a virtual machine object.

Aliases	none
Required?	true
Position?	named
Default Value	none



Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **CloudProtectionInfo[]**

## **Examples**

### **1: Get the applications installed on a virtual machine.**

The first command gets the virtual machine object named VM01 and stores the object in the \$VM variable.

The second command gets all applications installed on VM01.

```
PS C:\> $VM = Get-SCVirtualMachine "VM01"
```

```
PS C:\> $Apps = Get-SCApplication -VM $VM
```

---

# Get-SCApplicationDeployment

---

## Get-SCApplicationDeployment

Gets the applications that have been added to an application profile.

### Syntax

Parameter Set: AP

```
Get-SCApplicationDeployment -ApplicationProfile <ApplicationProfile> [-Name <String> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ID

```
Get-SCApplicationDeployment -ID <Guid> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCApplicationDeployment cmdlet gets the applications that have been added to an application profile.

### Parameters

#### **-ApplicationProfile<ApplicationProfile>**

Specifies an application profile object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

---

## **-ID<Guid>**

Specifies the numerical identifier (as a globally unique identifier, or GUID) for a specific object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **ApplicationDeployment**

## Examples

### 1: Get all application deployments associated with a specific application profile.

The first command gets the application profile object named SvcWebAppProfile01 and stores the object in the \$AppProfile variable.

The second command gets all of the application deployment objects for the application profile stored in \$AppProfile and stores the objects in the \$AppDeployment array.

The last command displays the application deployment objects stored in \$AppDeployment to the user.

```
PS C:\> $AppProfile = Get-SCApplicationProfile -Name "SvcWebAppProfile01"
```

```
PS C:\> $AppDeployment = Get-SCApplicationDeployment -ApplicationProfile $AppProfile
```

```
PS C:\> $AppDeployment
```

## Related topics

[Add-SCApplicationDeployment](#)

[Remove-SCApplicationDeployment](#)

[Set-SCApplicationDeployment](#)

---

# Get-SCApplicationHost

---

## Get-SCApplicationHost

Gets the application host for a service or for all services.

### Syntax

Parameter Set: All

```
Get-SCApplicationHost [-All] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: GetByName

```
Get-SCApplicationHost -Name <String> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: GetByService

```
Get-SCApplicationHost -Service <Service> [-Name <String> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ID

```
Get-SCApplicationHost [-ID <Guid> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCApplicationHost cmdlet gets the application host for a service or, when used with the All parameter, for all services.

### Parameters

#### -All

Retrieves a full list of all subordinate objects independent of the parent object. For example, the command Get-SCVirtualDiskDrive -All retrieves all virtual disk drive objects regardless of the virtual machine object or template object that each virtual disk drive object is associated with.

Aliases	none
Required?	false
Position?	named
Default Value	none

---

Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-ID<Guid>**

Specifies the numerical identifier (as a globally unique identifier, or GUID) for a specific object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Service<Service>**

Specifies a VMM service object.

Aliases	none
Required?	true
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **ApplicationHost**

## **Examples**

### **1: Get the application host for a service.**

The first command gets the service object named Service01 and stores the object in the \$Service variable.

The second command gets the application host for the service stored in \$Service and stores the variable in the \$ApplicationHost variable.

The last command displays the properties of the application host stored in \$ApplicationHost to the user.

```
PS C:\> $Service = Get-SCService -Name "Service01"
```

```
PS C:\> $ApplicationHost = Get-SCApplicationHost -Service $Service
```

```
PS C:\> $ApplicationHost
```

---

## Related topics

[Get-SCService](#)



---

# Get-SCApplicationHostTemplate

---

## Get-SCApplicationHostTemplate

Gets the application host template for a service template or for all service templates.

### Syntax

Parameter Set: All

```
Get-SCApplicationHostTemplate [-All] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: GetByName

```
Get-SCApplicationHostTemplate -Name <String> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: GetByServiceTemplate

```
Get-SCApplicationHostTemplate -ServiceTemplate <ServiceTemplate> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ID

```
Get-SCApplicationHostTemplate [-ID <Guid> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

## Detailed Description

The Get-SCApplicationHostTemplate gets the application host template for a service template or, when used with the All parameter, for all service templates.

## Parameters

### -All

Retrieves a full list of all subordinate objects independent of the parent object. For example, the command Get-SCVirtualDiskDrive -All retrieves all virtual disk drive objects regardless of the virtual machine object or template object that each virtual disk drive object is associated with.

Aliases	none
Required?	false
Position?	named

---

Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-ID<Guid>**

Specifies the numerical identifier (as a globally unique identifier, or GUID) for a specific object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-ServiceTemplate<ServiceTemplate>**

Specifies a service template object.

Aliases	none
Required?	true
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **ApplicationHostTemplate**

## **Examples**

### **1: Get the application host template for a specific service template.**

The first command gets the service template object named ServiceTemplate01 and stores the object in the \$ServiceTemplate variable.

The second command gets the application host template for the service template in \$ServiceTemplate and displays information about the application host template to the user.

```
PS C:\> $ServiceTemplate = Get-SCServiceTemplate -Name "ServiceTemplate01"
```

```
PS C:\> Get-SCApplicationHostTemplate -ServiceTemplate $ServiceTemplate
```

---

## Related topics

[Add-SCApplicationHostTemplate](#)

[Get-SCServiceTemplate](#)

[Remove-SCApplicationHostTemplate](#)

[Set-SCApplicationHostTemplate](#)

---

# Get-SCApplicationPackage

---

## Get-SCApplicationPackage

Gets an application package.

### Syntax

Parameter Set: All

```
Get-SCApplicationPackage [-All] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: EquivalentResourceParamSet

```
Get-SCApplicationPackage -FamilyName <String> [-Release <String> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ID

```
Get-SCApplicationPackage [-ID <Guid> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: NameParamSet

```
Get-SCApplicationPackage -Name <String> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

## Detailed Description

The Get-SCApplicationPackage cmdlet gets an application package from the VMM library.

## Parameters

### -All

Retrieves a full list of all subordinate objects independent of the parent object. For example, the command Get-SCVirtualDiskDrive -All retrieves all virtual disk drive objects regardless of the virtual machine object or template object that each virtual disk drive object is associated with.

Aliases	none
Required?	false
Position?	named
Default Value	none

---

Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-FamilyName<String>**

Specifies a family name for a physical resource in the VMM library. This value is used in conjunction with Release, Namespace, and Type to establish equivalency among library resources.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-ID<Guid>**

Specifies the numerical identifier (as a globally unique identifier, or GUID) for a specific object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	true
Position?	named

---

Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Release<String>**

Specifies a string that describes the release of a library resource. VMM automatically creates a release value for every resource imported into the library. After the resource has been imported, the string can be customized.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

---

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **ApplicationPackage**

## Examples

### 1: Get an application package by its name.

The first command gets the application package object named WebApp01.zip from the VMM library and stores the object in the \$AppPackage variable.

The second command displays information about the application package stored in \$AppPackage to the user.

```
PS C:\> $AppPackage = Get-SCApplicationPackage -Name "WebApp01.zip"
```

```
PS C:\> $AppPackage
```

## Related topics

[Set-SCApplicationPackage](#)



---

# Get-SCApplicationProfile

---

## Get-SCApplicationProfile

Gets application profiles.

### Syntax

Parameter Set: All

```
Get-SCApplicationProfile [-All] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: GetByApplicationHostTemplate

```
Get-SCApplicationProfile -ApplicationHostTemplate <ApplicationHostTemplate> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: GetByName

```
Get-SCApplicationProfile -Name <String> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: GetByVMTemplate

```
Get-SCApplicationProfile -VMTemplate <Template> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ID

```
Get-SCApplicationProfile [-ID <Guid> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

## Detailed Description

The Get-SCApplicationProfile cmdlet gets application profiles. You can get individual profiles by using parameters such as Name or ID, or get all application profiles in Virtual Machine Manager (VMM) by using the All parameter.

## Parameters

### -All

Retrieves a full list of all subordinate objects independent of the parent object. For example, the command Get-SCVirtualDiskDrive -All retrieves all virtual disk drive objects regardless of the virtual machine object or template object that each virtual disk drive object is associated with.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-ApplicationHostTemplate<ApplicationHostTemplate>**

Specifies an application host template object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-ID<Guid>**

Specifies the numerical identifier (as a globally unique identifier, or GUID) for a specific object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-VMTemplate<Template>**

Specifies a VMM template object used to create virtual machines.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

---

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **ApplicationProfile**

## Examples

### 1: Get an application profile by its name.

The first command gets the application profile object named SvcWebAppProfile01 and stores the object in the \$AppProfile variable.

The second command displays information about the application profile stored in \$AppProfile to the user.

```
PS C:\> $AppProfile = Get-SCApplicationProfile -Name "SvcWebAppProfile01"
```

```
PS C:\> $AppProfile
```

### 2: Get all application profiles.

The first command gets all application profile objects and stores them in the \$AppProfiles array.

The second command displays information about only the first object in the \$AppProfiles array to the user.

```
PS C:\> $AppProfiles = Get-SCApplicationProfile -All
```

```
PS C:\> $AppProfiles[0]
```

## Related topics

[New-SCApplicationProfile](#)

[Remove-SCApplicationProfile](#)

[Set-SCApplicationProfile](#)

---

# Get-SCApplicationSetting

---

## Get-SCApplicationSetting

Gets application settings for an application or application deployment.

### Syntax

Parameter Set: ApplicationDeployment

```
Get-SCApplicationSetting -ApplicationDeployment <ApplicationDeployment> [-Name <String> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: Application

```
Get-SCApplicationSetting -Application <SCApplication> [-Name <String> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ApplicationPackage

```
Get-SCApplicationSetting -ApplicationPackage <ApplicationPackage> [-Name <String> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

## Detailed Description

The Get-SCApplicationSetting cmdlet gets application settings for an application or application deployment.

## Parameters

### -Application<SCApplication>

Specifies an application object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

---

## **-ApplicationDeployment<ApplicationDeployment>**

Specifies an application deployment object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **-ApplicationPackage<ApplicationPackage>**

Specifies an application package object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **ApplicationSetting**

## **Examples**

### **1: Get all application settings for an application deployment.**

The first command gets the application profile object named SvcWebAppProfile01 and stores the object in the \$AppProfile variable.

The second command gets the application deployment object named SvcWebDeployment01 for the application profile stored in \$AppProfile and stores the object in the \$AppDeployment variable.

The last command gets the application setting objects for the application deployment stored in \$AppDeployment and stores the objects in the \$AppSetting array (this example assumes that there are multiple settings for the application).

```
PS C:\> $AppProfile = Get-SCApplicationProfile -Name "SvcWebAppProfile01"
```

```
PS C:\> $AppDeployment = Get-SCApplicationDeployment -ApplicationProfile $AppProfile -Name "SvcWebDeployment01"
```

```
PS C:\> $AppSetting = Get-SCApplicationSetting -ApplicationDeployment $AppDeployment
```

---

## 2: Get all application settings for an application installed on a virtual machine.

The first command gets the virtual machine object named VM01 and stores the object in the \$VM variable.

The second command gets the application objects installed on VM01 and stores the objects in the \$Apps variable.

The last command gets the application settings for the first application stored in \$Apps.

```
PS C:\> $VM = Get-SCVirtualMachine "VM01"
```

```
PS C:\> $Apps = Get-SCApplication -VM $VM
```

```
PS C:\> $AppSetting = Get-SCApplicationSetting -ApplicationDeployment $Apps[0]
```

### Related topics

[Get-SCApplication](#)

[Get-SCApplicationDeployment](#)

[Get-SCApplicationProfile](#)

[Set-SCApplicationSetting](#)



---

# Get-SCAvailabilitySetName

---

## Get-SCAvailabilitySetName

Gets the names of availability sets associated with virtual machines or computer tier configurations.

### Syntax

Parameter Set: Default

```
Get-SCAvailabilitySetName [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCAvailabilitySetName cmdlet gets the names of availability sets that are associated with virtual machines or computer tier configurations.

You can specify availability set names in a virtual machine configuration or a computer tier template.

### Parameters

#### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

#### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

---

## Outputs

The output type is the type of the objects that the cmdlet emits.

- `String[]`

## Examples

### **1: Get the names of all availability sets on a specified VMM management server.**

This command gets all the names of availability sets on VMMServer01.

```
PS C:\> Get-SCAvailabilitySetName -VMMServer "VMMServer01"
```

---

# Get-SCBaseline

---

## Get-SCBaseline

Gets one or more baselines objects.

### Syntax

Parameter Set: ID

```
Get-SCBaseline [-ID <Guid> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: Name

```
Get-SCBaseline [-Name <String> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCBaseline cmdlet gets one or more baseline objects. A baseline is a list of updates which, together with scope assignments, can grade the compliance of required updates for Virtual Machine Manager (VMM) fabric servers.

### Parameters

#### **-ID<Guid>**

Specifies the numerical identifier (as a globally unique identifier, or GUID) for a specific object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

#### **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **Baseline**

## **Examples**

### **1: Get a baseline by its name.**

This command gets the baseline object named Security Baseline.

```
PS C:\> Get-SCBaseline -Name "Security Baseline"
```

---

## Related topics

[New-SCBaseline](#)

[Remove-SCBaseline](#)

[Set-SCBaseline](#)

---

# Get-SCCapabilityProfile

---

## Get-SCCapabilityProfile

Gets a capability profile.

### Syntax

Parameter Set: FromName

```
Get-SCCapabilityProfile [-Name <String> ] [-VMMServer <ServerConnection> ] [ <CommonParameters> ]
```

### Detailed Description

The Get-SCCapabilityProfile cmdlet gets one or more capability profile objects in Virtual Machine Manager (VMM).

### Parameters

#### **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

#### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
---------	------

---

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **CloudCapabilityProfile**

## Examples

### 1: Get a capability profile by its name.

This command gets the capability profile object named CapabilityProf01 and displays information about the object to the user.

```
PS C:\> Get-SCCapabilityProfile -Name "CapabilityProf01"
```

### 1: Using OBO

This command gets the capability profile object named CapabilityProf01 and displays information about the object to the user.

```
PS C:\> $UR1 = Get-SCUserRole -Name UR1
```

```
PS C:\> Get-SCCapabilityProfile -Name "CapabilityProf01" -OnBehalfOfUserRole $UR1 -  
OnBehalfOfUser user1@bluecola.com
```

## Related topics

[New-SCCapabilityProfile](#)

[Remove-SCCapabilityProfile](#)

[Set-SCCapabilityProfile](#)

[Test-SCCapabilityProfile](#)

---

# Get-SCCertificate

---

## Get-SCCertificate

Gets a security certificate object from a VMware vCenter Server, from a VMware ESX host, or from a Citrix XenServer host.

### Syntax

Parameter Set: ByLocalFilename

```
Get-SCCertificate [-Filename] <String> [-VMMServer <ServerConnection> ] [ <CommonParameters> ]
```

Parameter Set: ByRemoteComputer

```
Get-SCCertificate [-ComputerName] <String> [-TCPPort <UInt32> ] [-VMMServer <ServerConnection> ] [ <CommonParameters> ]
```

### Detailed Description

The Get-SCCertificate cmdlet gets a security certificate object from a vCenter Server, from an ESX host, or from a XenServer host. You can use this cmdlet to import a non-trusted certificate into Virtual Machine Manager (VMM) so that you can use the certificate with the Add-SCVirtualizationManager cmdlet, the Set-SCVirtualizationManager cmdlet, the Add-SCVMHost cmdlet or the Set-SCVMHost cmdlet.

The certificate is required in order to establish an SSL connection between the VMM server and the vCenter Server, ESX host, or XenServer host.

### Parameters

#### **-ComputerName<String>**

Specifies the name of a computer that VMM can uniquely identify on your network. Valid formats are: FQDN, IPv4 or IPv6 address, or NetBIOS name.

NOTE: See the examples for a specific cmdlet to determine how that cmdlet specifies the computer name.

Aliases	none
Required?	true
Position?	1



---

Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Filename<String>**

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-TCPPort<UInt32>**

Specifies a numeric value that represents a TCP port.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named

---

Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **Certificate**

## Examples

### 1: Retrieve the security certificate for the specified VMware vCenter Server.

This command gets the security certificate object for the vCenter Server server named vCenterServer01 located in the Contoso.com domain, and displays the security certificate information.

```
PS C:\> Get-SCCertificate -Computername "vCenterServer01.Contoso.com"
```

### 2: Retrieve the security certificate for a specified VMware ESX host.

The first command gets the security certificate object from the ESX host named ESXHost01 and stores the object in the \$ESXCert variable.

The second command passes the contents of \$ESXCert to the Get-Member cmdlet, which displays the .NET type and a list of methods and properties for the certificate object.

```
PS C:\> $ESXCert = Get-SCCertificate -ComputerName "ESXHost01.Contoso.com"
```

```
PS C:\> $ESXCert | Get-Member
```

### 3: Retrieve the security certificate for the specified Citrix XenServer host.

This command gets the security certificate object for the XenServer named XenServer01 located in the Contoso.com domain, and displays the security certificate information.

```
PS C:\> Get-SCCertificate -Computername "XenServer01.Contoso.com"
```

---

## Related topics

[Add-SCVirtualizationManager](#)

[Add-SCVMHost](#)

[Set-SCVirtualizationManager](#)

[Set-SCVMHost](#)

---

# Get-SCCloud

---

## Get-SCCloud

Gets a private cloud object.

### Syntax

Parameter Set: Name

```
Get-SCCloud [[-Name] <String> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ID

```
Get-SCCloud -ID <Guid> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCCloud cmdlet gets a private cloud object from Virtual Machine Manager (VMM).

For more information about private clouds, type: "Get-Help New-SCCloud -detailed".

### Parameters

#### **-ID<Guid>**

Specifies the numerical identifier (as a globally unique identifier, or GUID) for a specific object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

#### **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	false
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Examples**

### **1: Get a specified private cloud.**

This command gets the private cloud object named Cloud01 on VMMServer01.

```
PS C:\> Get-SCCloud -VMMServer "VMMServer01.Contoso.com" -Name "Cloud01"
```

## **Related topics**

[New-SCCloud](#)

[Remove-SCCloud](#)

[Set-SCCloud](#)



---

# Get-SCCloudCapacity

---

## Get-SCCloudCapacity

Gets the cloud capacity for a private cloud.

### Syntax

Parameter Set: Default

```
Get-SCCloudCapacity [[-Cloud] <Cloud> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCCloudCapacity cmdlet gets the cloud capacity for a private cloud in Virtual Machine Manager (VMM). Cloud capacity includes settings for the number of virtual machines, number of virtual CPUs, custom quota points, storage, and memory assigned to a private cloud.

### Parameters

#### **-Cloud<Cloud>**

Specifies a private cloud object.

Aliases	none
Required?	false
Position?	1
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

#### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

---

## Examples

### 1: Get the cloud capacity for a specified private cloud.

The first command gets the private cloud object named Cloud01 and stores the object in the \$Cloud variable.

The second command gets the cloud capacity for the private cloud object stored in \$Cloud, and displays information about the cloud capacity to the user.

```
PS C:\> $Cloud = Get-SCCloud -Name "Cloud01"
```

```
PS C:\> Get-SCCloudCapacity -Cloud $Cloud
```

## Related topics

[Get-SCCloud](#)

[Set-SCCloud](#)

[Set-SCCloudCapacity](#)



---

# Get-SCCloudProtectionInfo

---

## Get-SCCloudProtectionInfo

This cmdlet is reserved for future use.

### Syntax

Parameter Set: Default

```
Get-SCCloudProtectionInfo [-Cloud] <Cloud> [-PrimaryCloudInfo] [-RecoveryCloudInfo] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

This cmdlet is reserved for future use.

### Parameters

#### -Cloud<Cloud>

Specifies a private cloud object.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

#### -PrimaryCloudInfo

This parameter is reserved for future use.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-RecoveryCloudInfo**

This parameter is reserved for future use.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

---

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **CloudProtectionInfo[]**

## Examples

PS C:\> # This cmdlet is reserved for future use.

---

# Get-SCCloudUsage

---

## Get-SCCloudUsage

Gets cloud usage data for a specified private cloud in VMM.

### Syntax

Parameter Set: CloudUsage

```
Get-SCCloudUsage -Cloud <Cloud> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: UserRoleUsageOfCloud

```
Get-SCCloudUsage -Cloud <Cloud> -UserRole <UserRole> [-UserName <String> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCCloudUsage cmdlet gets cloud usage data for a specified private cloud in Virtual Machine Manager (VMM). Cloud usage data includes the following:

- Number of CPUs
- Custom quota points
- Amount of memory (in MB)
- Amount of storage (in GB)
- Number of Virtual Machines

You can scope this data to usage per user role and per user.

### Parameters

#### **-Cloud<Cloud>**

Specifies a private cloud object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)

---

Accept Wildcard Characters?	false
-----------------------------	-------

### **-UserName<String>**

Specifies a the name of a user. Enter a user name with the format Domain\User.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-UserRole<UserRole>**

Specifies a user role object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)

---

Accept Wildcard Characters?
-----------------------------

false
-------

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Examples

### 1: Get the usage information for a specified private cloud.

The first command gets the private cloud object named Cloud01 and stores the object in the \$Cloud variable.

The second command gets the cloud usage information for the private cloud stored in \$Cloud and displays the following information to the user:

- CPUUsageCount
- CustomQuotaUsageCount
- MemoryUsageMB
- StorageUsageGB
- VMUsageCount

```
PS C:\> $Cloud = Get-SCCloud -Name "Cloud01"
```

```
PS C:\> Get-SCCloudUsage -Cloud $Cloud
```

### 2: Get the usage information for a specified private cloud for a specified user role.

The first command gets the private cloud object named Cloud01 and stores the object in the \$Cloud variable.

The second command gets the user role object named SelfServiceUsers and stores the object in the \$UserRole variable.

The last command gets the cloud usage information for the private cloud stored in \$Cloud and the user role stored in \$UserRole. Then, the command displays the cloud usage information to the user.

```
PS C:\> $Cloud = Get-SCCloud -Name "Cloud01"
```

```
PS C:\> $UserRole = Get-SCUserRole -Name "SelfServiceUsers"
```

```
PS C:\> Get-SCCloudUsage -Cloud $Cloud -UserRole $UserRole
```

---

### 3: Get the usage information for a specified private cloud for a specified user within a user role.

The first command gets the private cloud object named Cloud01 and stores the object in the \$Cloud variable.

The second command gets the user role object named SelfServiceUsers and stores the object in the \$UserRole variable.

The last command gets the cloud usage information for the private cloud stored in \$Cloud and the user named Katarina who is in the user role named SelfServiceUsers. Then, the command displays the cloud usage information to the user.

```
PS C:\> $Cloud = Get-SCCloud -Name "Cloud01"
```

```
PS C:\> $UserRole = Get-SCUserRole -Name "SelfServiceUsers"
```

```
PS C:\> Get-SCCloudUsage -Cloud $Cloud -UserRole $UserRole -UserName "Contoso\Katarina"
```

### Related topics

[Get-SCUserRole](#)

[Get-SCCloud](#)

---

# Get-SCClusterVirtualNetwork

---

## Get-SCClusterVirtualNetwork

Gets the virtual network associated with a host cluster.

### Syntax

Parameter Set: Default

```
Get-SCClusterVirtualNetwork [-VMHostCluster] <HostCluster> [-VMMServer <ServerConnection> ]  
[ <CommonParameters>]
```

### Detailed Description

The Get-SCClusterVirtualNetwork cmdlet gets one or more virtual networks associated with a host cluster.

### Parameters

#### **-VMHostCluster<HostCluster>**

Specifies a VMM host cluster object.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

#### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
---------	------



---

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **VirtualNetwork**

## Examples

### 1: Get the virtual network for a cluster.

The first command gets the cluster object named VMHostCluster01 and stores the object in the \$Cluster variable.

The second command gets the cluster virtual network object for the cluster stored in \$Cluster (VMHostCluster01).

```
PS C:\> $Cluster = Get-SCVMHostCluster -Name "VMHostCluster01.Contoso.com"
```

```
PS C:\> Get-SCClusterVirtualNetwork -VMHostCluster $Cluster
```

## Related topics

[Get-SCVMHostCluster](#)

---

# Get-SCComplianceStatus

---

## Get-SCComplianceStatus

Gets the compliance status of computers managed by VMM.

### Syntax

Parameter Set: All

```
Get-SCComplianceStatus [-All] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: VMMManagedComputer

```
Get-SCComplianceStatus -VMMManagedComputer <VMMManagedComputer> [-VMMServer  
<ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCComplianceStatus cmdlet gets the compliance status object of one or more computers managed by Virtual Machine Manager (VMM). The returned compliance status provides details on the compliance against assigned baselines.

### Parameters

#### -All

Retrieves a full list of all subordinate objects independent of the parent object. For example, the command Get-SCVirtualDiskDrive -All retrieves all virtual disk drive objects regardless of the virtual machine object or template object that each virtual disk drive object is associated with.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-VMMManagedComputer<VMMManagedComputer>**

Specifies a computer object that is managed by VMM.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **ComplianceStatus**

---

## Examples

### 1: Get the compliance status of a host.

The first command gets the host object named VMHost01 and stores the object in the \$VMHost variable.

The second command gets the compliance status object for the host stored in \$VMHost01 and stores the object in the \$Compliance variable.

The last command displays information about the compliance status of VMHost01 to the user.

```
PS C:\> $VMHost = Get-SCVMHost -ComputerName "VMHost01"
```

```
PS C:\> $Compliance = Get-SCComplianceStatus -VMManagedComputer $VMHost.ManagedComputer
```

```
PS C:\> $Compliance
```

## Related topics

[Set-SCComplianceStatus](#)

[Start-SCComplianceScan](#)

---

# Get-SCComputerConfiguration

---

## Get-SCComputerConfiguration

Gets physical machine configuration objects from the VMM database.

### Syntax

Parameter Set: Default

```
Get-SCComputerConfiguration [-SourceComputerName <String> ] [-VMMServer <ServerConnection> ]  
[ <CommonParameters>]
```

### Detailed Description

The Get-SCComputerConfiguration cmdlet gets one or more computer configuration objects from the Virtual Machine Manager (VMM) database that are associated with one or more physical computers. Information about a computer's hardware, physical disks, and operating system is stored in the machine configuration object.

A physical machine configuration is used by the New-SCP2V cmdlet when it converts a physical machine to a virtual machine. To perform this conversion, you use a physical computer as a model from which to create an identical, or nearly identical, virtual machine that has the same identity (ComputerName.DomainName) as the physical machine.

### Parameters

#### **-SourceComputerName<String>**

Specifies the source computer for a physical-to-virtual (P2V) machine conversion performed by VMM. Valid formats: FQDN, IPv4 or IPv6 address, or NetBIOS name.

Note: See the examples for a specific cmdlet to determine how that cmdlet specifies the source computer name.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false

Accept Wildcard Characters?	false
-----------------------------	-------

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **ComputerConfig**

## **Examples**

### **1: Get the computer configuration object for a particular physical machine.**

This command gets the machine configuration for the physical machine named P2VSource01.Contoso.com from the VMM database and displays information about this object to the user.

```
PS C:\> Get-SCComputerConfiguration | where { $_.Name -eq "P2VSource01.Contoso.com" }
```

### **2: Get all computer configuration objects in your VMM environment.**

This command gets all existing machine configuration objects on VMMServer01 and displays information about these machine configuration objects to the user.

```
PS C:\> Get-MachineConfig -VMMServer "VMMServer01.Contoso.com"
```

---

## Related topics

[New-SCComputerConfiguration](#)

[New-SCP2V](#)

[Remove-SCComputerConfiguration](#)

---

# Get-SCComputerTier

---

## Get-SCComputerTier

Gets a VMM computer tier object.

### Syntax

Parameter Set: All

```
Get-SCComputerTier [-All] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ID

```
Get-SCComputerTier [-ID <Guid> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: Service

```
Get-SCComputerTier -Service <Service> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCComputerTier cmdlet gets one or more Virtual Machine Manager (VMM) Computer Tier objects.

### Parameters

#### -All

Retrieves a full list of all subordinate objects independent of the parent object. For example, the command Get-SCVirtualDiskDrive -All retrieves all virtual disk drive objects regardless of the virtual machine object or template object that each virtual disk drive object is associated with.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false



---

## **-ID<Guid>**

Specifies the numerical identifier (as a globally unique identifier, or GUID) for a specific object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Service<Service>**

Specifies a VMM service object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **ComputerTier**

## Examples

### 1: Get a computer tier for a specific service.

The first command gets the service object named Service01 and stores the object in the \$Service variable.

The second command gets the computer tier for the service stored in \$Service.

The last command displays the properties of the computer tier stored in \$ComputerTier to the user.

```
PS C:\> $Service = Get-SCService -Name "Service01"
```

```
PS C:\> $ComputerTier = Get-SCComputerTier -Service $Service
```

```
PS C:\> $ComputerTier
```

## Related topics

[Set-SCComputerTier](#)

---

# Get-SCComputerTierConfiguration

---

## Get-SCComputerTierConfiguration

Gets the computer tier configuration for a service deployment configuration.

### Syntax

Parameter Set: Default

```
Get-SCComputerTierConfiguration -ServiceConfiguration <ServiceConfiguration> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCComputerTierConfiguration cmdlet gets the computer tier configuration for a service deployment configuration.

### Parameters

#### **-ServiceConfiguration<ServiceConfiguration>**

Specifies a service configuration object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

#### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
---------	------

---

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **ComputerTierConfiguration**

## Examples

### 1: Get the computer tier configuration object for a service configuration.

The first command gets the service configuration object named Service01 and stores the object in the \$ServiceConfig variable.

The second command gets the computer tier configuration for the service configuration stored in Service01 and stores the object in the \$TierConfig variable.

The last command displays the properties of the computer tier configuration stored in \$TierConfig to the user.

```
PS C:\> $ServiceConfig = Get-SCServiceConfiguration -Name "Service01"
```

```
PS C:\> $TierConfig = Get-SCComputerTierConfiguration -ServiceConfiguration $ServiceConfig
```

```
PS C:\> $TierConfig
```

## Related topics

[Get-SCServiceConfiguration](#)

[Set-SCComputerTierConfiguration](#)

---

# Get-SCComputerTierTemplate

---

## Get-SCComputerTierTemplate

Gets the computer tier template for a service template.

### Syntax

Parameter Set: Name

```
Get-SCComputerTierTemplate [-Name <String> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: All

```
Get-SCComputerTierTemplate [-All] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ID

```
Get-SCComputerTierTemplate [-ID <Guid> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ServiceTemplate

```
Get-SCComputerTierTemplate -ServiceTemplate <ServiceTemplate> [-Name <String> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCComputerTierTemplate cmdlet gets the computer tier template for a service template.

### Parameters

#### -All

Retrieves a full list of all subordinate objects independent of the parent object. For example, the command Get-SCVirtualDiskDrive -All retrieves all virtual disk drive objects regardless of the virtual machine object or template object that each virtual disk drive object is associated with.

Aliases	none
Required?	false
Position?	named
Default Value	none

---

Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-ID<Guid>**

Specifies the numerical identifier (as a globally unique identifier, or GUID) for a specific object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-ServiceTemplate<ServiceTemplate>**

Specifies a service template object.

Aliases	none
Required?	true
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **ComputerTierTemplate**

## **Examples**

### **1: Get the computer tier template for a specified service template.**

The first command gets the service template object named ServiceTemplate01 and stores the object in the \$ServiceTemplate variable.

The second command gets the computer tier template for the service template stored in \$ServiceTemplate.

The last command displays the properties of the computer tier template for the user.

```
PS C:\> $ServiceTemplate = Get-SCServiceTemplate -Name "ServiceTemplate01"
```

```
PS C:\> $TierTemplate = Get-SCComputerTierTemplate -ServiceTemplate $ServiceTemplate
```

```
PS C:\> $TierTemplate
```

---

## Related topics

[Add-SCComputerTierTemplate](#)

[Get-SCServiceTemplate](#)

[Remove-SCComputerTierTemplate](#)

[Set-SCComputerTierTemplate](#)



---

# Get-SCConfigurationProvider

---

## Get-SCConfigurationProvider

Gets a configuration provider object.

### Syntax

Parameter Set: Default

```
Get-SCConfigurationProvider [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ByName

```
Get-SCConfigurationProvider [[-Name] <String> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ByType

```
Get-SCConfigurationProvider [[-ProviderType] {OutOfBandManagement | LoadBalancer | Gateway | VirtualSwitchExtensionManager | Unknown} ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCConfigurationProvider cmdlet gets a configuration provider object. A configuration provider is a plug-in to Virtual Machine Manager (VMM) that translates VMM PowerShell commands to API calls that are specific to a type of load balancer or baseboard management controller.

### Parameters

#### **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	false
Position?	1
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

---

## **-ProviderType<ConfigurationProviderType>**

Specifies the type of a configuration provider. Valid values are: LoadBalancer, OutOfBandManagement. The acceptable values for this parameter are:

OutOfBandManagement	
LoadBalancer	
Gateway	
VirtualSwitchExtensionManager	
Unknown	

Aliases	none
Required?	false
Position?	1
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **ConfigurationProvider**

## Examples

### 1: Get all configuration providers by a specified type.

This command gets all configuration providers that are load balancers.

```
PS C:\> Get-SCConfigurationProvider -ProviderType "LoadBalancer"
```

### 2: Get a configuration provider by its name.

The first command gets the Microsoft Network Load Balancing (NLB) configuration provider by its name and stores it in the \$ConfigProvider variable.

The second command displays information about the configuration provider stored in \$ConfigProvider to the user.

```
PS C:\> $ConfigProvider = Get-SCConfigurationProvider -Name "Microsoft Network Load  
Balancing (NLB)"
```

```
PS C:\> $ConfigProvider
```

---

# Get-SCCPUType

---

## Get-SCCPUType

Gets CPU object types for use in virtual machines, or for use in templates or hardware profiles used to create virtual machines.

### Syntax

Parameter Set: FromGuid

```
Get-SCCPUType [-ID <Guid> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCCPUType cmdlet gets types of CPU objects that are available for use in virtual machines, or for use in templates or hardware profiles used to create virtual machines, in a Virtual Machine Manager (VMM) environment. The type of CPU is one of the factors that the VMM placement process uses to determine which hosts (among all available hosts) are suitable for a specific virtual machine.

### Parameters

#### **-ID<Guid>**

Specifies the numerical identifier (as a globally unique identifier, or GUID) for a specific object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

#### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

---

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **ProcessorType**

## Examples

### 1: Get all available processor types.

This command gets all available CPU object types on VMMServer01, formats the information about each CPU type in a list, and then displays this information to the user.

```
PS C:\> Get-SCCPUType -VMMServer "VMMServer01.Contoso.com"
```

## Related topics

[New-SCHardwareProfile](#)

[New-SCVirtualMachine](#)

[New-SCVMTemplate](#)

---

# Get-SCCustomPlacementRule

---

## Get-SCCustomPlacementRule

Gets the custom placement rules for a placement configuration.

### Syntax

Parameter Set: FromPlacementConfiguration

```
Get-SCCustomPlacementRule -PlacementConfiguration <PlacementConfigurationSettings> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCCustomPlacementRule cmdlet gets the custom placement rules that have been added to the placement configuration for a host group.

### Parameters

#### **-PlacementConfiguration<PlacementConfigurationSettings>**

Specifies a placement configuration object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

#### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
---------	------

---

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **CustomPlacementRule**

## Examples

### 1: Get all of the custom placement rules for a specified host group.

The first command gets the host group object named HostGroup01 and stores the object in the \$HostGroup variable.

The second command gets the placement configuration object for the host group stored in \$HostGroup and places the object in the \$PlacementConfig variable.

The third command gets all custom placement rule objects for the placement configuration stored in \$PlacementConfig and stores the objects in the \$CustomPlacementRule variable.

The last command displays information about the custom placement rules stored in \$CustomPlacementRule for the user.

```
PS C:\> $HostGroup = Get-SCVMHostGroup "HostGroup01"
```

```
PS C:\> $PlacementConfig = Get-SCPlacementConfiguration -VMHostGroup $HostGroup
```

```
PS C:\> $CustomPlacementRule = Get-SCCustomPlacementRule -PlacementConfiguration  
$PlacementConfig
```

```
PS C:\> $CustomPlacementRule
```

## Related topics

[Get-SCPlacementConfiguration](#)

[Get-SCVMHostGroup](#)

[Remove-SCCustomPlacementRule](#)

---

# Get-SCCustomProperty

---

## Get-SCCustomProperty

Gets a custom property definition from the VMM database.

### Syntax

Parameter Set: ID

```
Get-SCCustomProperty [-ID <Guid> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: Member

```
Get-SCCustomProperty -Member {VM | VMHost | Template | ServiceTemplate | ServiceInstance | Cloud | VMHostGroup | HostCluster | ComputerTier} [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: Name

```
Get-SCCustomProperty -Name <String> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

## Detailed Description

The Get-SCCustomProperty cmdlet gets a custom property definition from the Virtual Machine Manager (VMM) database.

## Parameters

### -ID<Guid>

Specifies the numerical identifier (as a globally unique identifier, or GUID) for a specific object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false



---

## **-Member<CustomPropertyObjectType>**

Specifies an object that is part of a group.

The acceptable values for this parameter are:

VM	
VMHost	
Template	
ServiceTemplate	
ServiceInstance	
Cloud	
VMHostGroup	
HostCluster	
ComputerTier	

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

### **Examples**

#### **1: Get a custom property by its name.**

The first command gets the custom property object named Cost Center and stores the object in the \$CustomProp variable.

The second command displays the properties of the custom property object stored in \$CustomProp to the user.

```
PS C:\> $CustomProp = Get-SCCustomProperty -Name "Cost Center"
```

```
PS C:\> $CustomProp
```

#### **2: Get all custom properties for a specific member type.**

This command returns all custom properties that contain VM as a member.

```
PS C:\> Get-SCCustomProperty -Member "VM"
```

### **Related topics**

[New-SCCustomProperty](#)

[Remove-SCCustomProperty](#)

[Set-SCCustomProperty](#)

---

# Get-SCCustomPropertyValue

---

## Get-SCCustomPropertyValue

Gets a custom property value object.

### Syntax

Parameter Set: All

```
Get-SCCustomPropertyValue -All [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: Filter

```
Get-SCCustomPropertyValue -CustomProperty <CustomProperty> -InputObject <ClientObject> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCCustomPropertyValue cmdlet gets a custom property value object.

### Parameters

#### -All

Retrieves a full list of all subordinate objects independent of the parent object. For example, the command Get-SCVirtualDiskDrive -All retrieves all virtual disk drive objects regardless of the virtual machine object or template object that each virtual disk drive object is associated with.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-CustomProperty<CustomProperty>**

Specifies a custom property object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-InputObject<ClientObject>**

Specifies the object that is assigned the property whose value you want to retrieve or change.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Examples

### 1: Get the value for a custom property on a virtual machine.

The first command gets the virtual machine object named VM01 and stores the object in the \$VM variable.

The second command gets the custom property object named Cost Center and stores the object in the \$CustomProp variable.

The last command retrieves the value for the custom property stored in \$CustomProp (Cost Center) for the virtual machine stored in \$VM (VM01).

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM01"
```

```
PS C:\> $CustomProp = Get-SCCustomProperty -Name "Cost Center"
```

```
PS C:\> Get-SCCustomPropertyValue -InputObject $VM -CustomProperty $CustomProp
```

## Related topics

[Get-SCCustomProperty](#)

[Remove-SCCustomPropertyValue](#)

[Set-SCCustomPropertyValue](#)

---

# Get-SCCustomResource

---

## Get-SCCustomResource

Gets a custom resource from the VMM library.

### Syntax

Parameter Set: All

```
Get-SCCustomResource [-All] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: EquivalentResourceParamSet

```
Get-SCCustomResource -FamilyName <String> [-Release <String> ] [-VMMServer  
<ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ID

```
Get-SCCustomResource [-ID <Guid> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: NameParamSet

```
Get-SCCustomResource -Name <String> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCCustomResource cmdlet gets a custom resource from the VMM library. A custom resource is a folder-based library object in Virtual Machine Manager (VMM). The resource is declared at the folder level, and the contents of the folder is unknown to VMM.

To add a custom resource to the library, create a folder with a .CR extension, place content in the folder, and then use the VMM console to drag the folder to a VMM library share. VMM discovers and imports the folder into the library as a custom resource.

### Parameters

#### -All

Retrieves a full list of all subordinate objects independent of the parent object. For example, the command Get-SCVirtualDiskDrive -All retrieves all virtual disk drive objects regardless of the virtual machine object or template object that each virtual disk drive object is associated with.

Aliases	none
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-FamilyName<String>**

Specifies a family name for a physical resource in the VMM library. This value is used in conjunction with Release, Namespace, and Type to establish equivalency among library resources.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-ID<Guid>**

Specifies the numerical identifier (as a globally unique identifier, or GUID) for a specific object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
---------	------

Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Release<String>**

Specifies a string that describes the release of a library resource. VMM automatically creates a release value for every resource imported into the library. After the resource has been imported, the string can be customized.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).



---

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **CustomResource**

## Examples

### 1: Get a specific custom resource.

This command gets the custom resource object named Folder.CR on LibraryServer01 from the VMM library on VMMServer01 and then stores the object in the \$CR variable.

```
PS C:\> $CR = Get-SCCustomResource -VMMServer "VMMServer01.Contoso.com" | where { $_.Name -eq "Folder.CR" -and $_.LibraryServer.Name -eq "LibraryServer01.Contoso.com" }
```

## Related topics

[Remove-SCCustomResource](#)

[Set-SCCustomResource](#)

---

# Get-SCDependentLibraryResource

---

## Get-SCDependentLibraryResource

Identifies dependencies between VMM objects.

### Syntax

Parameter Set: LibraryResource

```
Get-SCDependentLibraryResource [-LibraryResource] <LibObjectBase> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: LibraryServer

```
Get-SCDependentLibraryResource [-LibraryServer] <LibraryServer> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: LibraryShare

```
Get-SCDependentLibraryResource [-LibraryShare] <LibraryShare> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCDependentLibraryResource cmdlet identifies dependencies between Virtual Machine Manager (VMM) objects.

You can use the Get-SCDependentLibraryResource cmdlet to identify objects that are dependent on the existence of:

- The specified library object
- Any object on the specified library share
- Any object on the specified library server

### Parameters

#### **-LibraryResource<LibObjectBase>**

Specifies a resource stored in the VMM library.

Aliases	none
Required?	true

---

Position?	1
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-LibraryServer<LibraryServer>**

Specifies a VMM library server object.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-LibraryShare<LibraryShare>**

Specifies a VMM library share object.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false

---

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **HardwareProfile**
- **VirtualMachine**
- **Template**
- **GuestOSProfile**

## Examples

### 1: Find all objects that depend on a particular virtual hard disk.

The first command gets the hard disk object named VHD01 on LibraryServer01 from the VMM library on VMMServer01 and stores the object in the \$VHD variable. This example assumes that only one virtual hard disk named VHD01 exists.

The second command returns all of the library objects that are dependent on VHD01.

If dependent objects exist, removing this virtual hard disk will modify those dependent objects so that they no longer reference the removed virtual hard disk. Thus, if VHD01 is associated with a specific virtual machine or with a specific template, that virtual machine or template is modified so that it no longer references VHD01 after it is removed.

```
PS C:\> $VHD = Get-SCVirtualHardDisk -VMMServer "VMMServer01.Contoso.com" | where { $_.Name -eq "VHD01" -and $_.LibraryServer.Name -eq "LibraryServer01.Contoso.com" }
```

```
PS C:\> Get-SCDependentLibraryResource -LibraryResource $VHD
```

## Related topics

[Get-SCISO](#)

[Get-SCLibraryServer](#)

[Get-SCLibraryShare](#)

[Get-SCVirtualHardDisk](#)

---

[Get-SCVirtualMachine](#)

[Get-SCVMTemplate](#)

---

# Get-SCDirectoryChildItem

---

## Get-SCDirectoryChildItem

Gets all files and subdirectories in the specified directory on a virtual machine host or on a library server managed by VMM.

### Syntax

Parameter Set: FromLibraryServer

```
Get-SCDirectoryChildItem -LibraryServer <LibraryServer> -Path <String> [ <CommonParameters>]
```

Parameter Set: FromVMHost

```
Get-SCDirectoryChildItem -Path <String> -VMHost <Host> [ <CommonParameters>]
```

### Detailed Description

The Get-SCDirectoryChildItem gets all files and subdirectories immediately under the specified directory on a virtual machine host or on a library server managed by Virtual Machine Manager (VMM). If you specify a share path (such as \\ServerName\ShareName\Directory\FileName), the subdirectories of the share path are returned.

If you use the Get-SCDirectoryChildItem cmdlet to retrieve files and subdirectories on a library server, you must specify a path to a valid library share. For example, the share path to the default library share installed by Setup when you first install VMM is:

```
\\VMMServerName.DomainName.com\MSSCVMMLibrary
```

### Parameters

#### **-LibraryServer<LibraryServer>**

Specifies a VMM library server object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)

Accept Wildcard Characters?	false
-----------------------------	-------

## **-Path<String>**

Specifies the destination path for the operation.

Example formats:

Local path -Path "F:\"

UNC path -Path "\\Library\Templates"

Volume GUID path -Path "\\?\Volume{4703c1ea-8ae7-11db-b473-00123f7603e3}\"

VMware ESX path -Path "[storage1]\MyVMwareFolderForVMs\MyVM.vmx"

Citrix XenServer path - Path "Local storage[99b6212f-b63d-c676-25f9-d6c460992de7]"

Wildcards are supported for "Get" cmdlets and when you specify the UNC path:

Example format:

UNC path -Path "\\VMHostServer\MyVMs\\*VM\*"

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMHost<Host>**

Specifies a virtual machine host object. VMM supports Hyper-V hosts, VMware ESX hosts, and Citrix XenServer hosts.

For more information about each type of host, type: "Get-Help Add-SCVMHost -detailed". See the examples for a specific cmdlet to determine how that cmdlet uses this parameter.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **VMHostFileInformation**

## Examples

### 1: Get the files and subdirectories for the specified path on a VMM host.

The first command gets the host object named VMHost01 from the VMM database and stores the object in the \$VMHost variable.

The second command displays the name and other information about each file and subdirectory immediately under the C:\ drive on VMHost01.

```
PS C:\> $VMHost = Get-SCVMHost -ComputerName "VMHost01"
```

```
PS C:\> Get-SCDirectoryChildItem -VMHost $VMHost -Path "C:\"
```

### 2: Get the subdirectories for the specified path on a library server.

The first command gets the library server object named FileServer01 from VMMServer01 and stores the object in the \$LibServ variable.

The second command displays the name, parent directory, and other information about each file stored in the directory for the default library share on FileServer01. You must specify the complete path to the library share.

NOTE: This example assumes that the default VMM library share (MSSCVMLibrary) is used in your environment. To get the names of library shares, type: "Get-SCLibraryShare | select Name".

```
PS C:\> $LibServ = Get-SCLibraryServer -VMMServer "VMMServer01.Contoso.com" -ComputerName "FileServer01.Contoso.com"
```

```
PS C:\> Get-SCDirectoryChildItem -LibraryServer $LibServ -Path "\\FileServer01.Contoso.com\MSSCVMLibrary"
```

## Related topics

[Get-SCLibraryServer](#)

[Get-SCVMHost](#)



---

# Get-SCDriverPackage

---

## Get-SCDriverPackage

Gets a driver package stored in a VMM library share.

### Syntax

Parameter Set: Connection

```
Get-SCDriverPackage [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: All

```
Get-SCDriverPackage [-All] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ID

```
Get-SCDriverPackage [-ID <Guid> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: PNPIdMatching

```
Get-SCDriverPackage [[-PnPID] <List<String>> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: TagMatching

```
Get-SCDriverPackage [[-Tag] <List<String>> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

## Detailed Description

The Get-SCDriverPackage cmdlet gets one or more driver packages stored in a Virtual Machine Manager (VMM) library share.

## Parameters

### -All

Retrieves a full list of all subordinate objects independent of the parent object. For example, the command Get-SCVirtualDiskDrive -All retrieves all virtual disk drive objects regardless of the virtual machine object or template object that each virtual disk drive object is associated with.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-ID<Guid>**

Specifies the numerical identifier (as a globally unique identifier, or GUID) for a specific object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-PnPID<List<String>>**

Specifies the plug and play (PnP) ID that the driver package supports. You can use this parameter to query for driver packages with matching PnP ID values.

Aliases	none
Required?	false
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Tag<List<String>>**

Specifies a word or phrase to associate with an object so that you can search for all objects with the specified set of tags. You can search for a subset of tags, or you can search for the full set of tags.

Aliases	none
Required?	false
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **DriverPackage[]**
- **DriverPackage**

## **Examples**

### **1: Get all driver packages stored in the VMM library.**

This command gets all driver packages stored in the VMM library and displays information about each driver package to the user.

```
PS C:\> Get-SCDriverPackage
```

---

## 2: Get a driver package by its Plug and Play ID.

This command gets the driver package that has a Plug and Play ID (PnPID) of "ROOT\VMBUS".

```
PS C:\> Get-SCDriverPackage -PnPID "ROOT\VMBUS"
```

## 3: Get all driver packages with a specified tag value.

This command gets all driver packages that have a tag value of Production.

```
PS C:\> Get-SCDriverPackage -Tag "Production"
```

## Related topics

[Remove-SCDriverPackage](#)

[Set-SCDriverPackage](#)

---

# Get-SCDynamicOptimizationConfiguration

---

## Get-SCDynamicOptimizationConfiguration

Gets the dynamic optimization configuration for a host group.

### Syntax

Parameter Set: FromHostGroup

```
Get-SCDynamicOptimizationConfiguration -VMHostGroup <HostGroup> [-VMMServer <ServerConnection> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The Get-SCDynamicOptimizationConfiguration cmdlet gets the dynamic optimization configuration for a host group.

### Parameters

#### **-VMHostGroup<HostGroup>**

Specifies a virtual machine host group object or an array of host group objects.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

#### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
---------	------

---

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Confirm**

Prompts you for confirmation before executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **DynamicOptimizationConfiguration**

---

## Examples

### 1: Get the Dynamic Optimization settings for a host group.

The first command gets the host group object named HostGroup01 and stores the object in the \$HostGroup variable.

The second command gets the dynamic optimization configuration settings for the host group stored in \$HostGroup and stores the settings in the \$DOConfig variable.

The last command displays the settings stored in \$DOConfig to the user.

```
PS C:\> $HostGroup = Get-SCVMHostGroup "HostGroup01"
```

```
PS C:\> $DOConfig = Get-SCDynamicOptimizationConfiguration -VMHostGroup $HostGroup
```

```
PS C:\> $DOConfig
```

## Related topics

[Get-SCVMHostGroup](#)

[Set-SCDynamicOptimizationConfiguration](#)

---

# Get-SCExtensionUplinkPortProfile

---

## Get-SCExtensionUplinkPortProfile

Gets an extension uplink port profile.

### Syntax

Parameter Set: All

```
Get-SCExtensionUplinkPortProfile [[-Name] <String> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ByID

```
Get-SCExtensionUplinkPortProfile [[-Name] <String> ] -ID <Guid> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ByVirtualSwitchExtension

```
Get-SCExtensionUplinkPortProfile [[-Name] <String> ] -VirtualSwitchExtension <VirtualSwitchExtension> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ByVirtualSwitchExtensionManager

```
Get-SCExtensionUplinkPortProfile [[-Name] <String> ] -VirtualSwitchExtensionManager <VirtualSwitchExtensionManager> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCExtensionUplinkPortProfile cmdlet gets one or more extension uplink port profiles. You can get an extension uplink port profile by its name or ID, or by its virtual switch extension or virtual switch extension manager.

### Parameters

#### **-ID<Guid>**

Specifies the numerical identifier (as a globally unique identifier, or GUID) for a specific object.

Aliases	none
Required?	true
Position?	named



Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	false
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VirtualSwitchExtension<VirtualSwitchExtension>**

Specifies a virtual switch extension object.

To get a virtual switch extension object, use the Get-SCVirtualSwitchExtension cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VirtualSwitchExtensionManager<VirtualSwitchExtensionManager>**

Specifies a virtual switch extension manager object.

To obtain a virtual switch extension manager object, use the Get-SCVirtualSwitchExtensionManager cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Related topics**

[Get-SCVirtualSwitchExtension](#)

[Get-SCVirtualSwitchExtensionManager](#)

---

# Get-SCGatewayConnection

---

## Get-SCGatewayConnection

Gets a gateway connection.

### Syntax

Parameter Set: All

```
Get-SCGatewayConnection [-All] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCGatewayConnection cmdlet gets a gateway connection. You can get all gateway connections, or gateway connections for a specified VMM server.

### Parameters

#### -All

Retrieves a full list of all subordinate objects independent of the parent object. For example, the command Get-SCVirtualDiskDrive -All retrieves all virtual disk drive objects regardless of the virtual machine object or template object that each virtual disk drive object is associated with.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

#### -VMMServer<ServerConnection>

Specifies a VMM server object.

---

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Examples

### 1: Get all gateway connections.

This command gets all gateway connections and displays information about each connection to the user.

```
PS C:\> Get-SCGatewayConnection -All
```

## Related topics

[Get-SCVMMServer](#)

---

# Get-SCGuestOSProfile

---

## Get-SCGuestOSProfile

Gets a guest operating system profile object from the VMM library.

### Syntax

Parameter Set: Connection

```
Get-SCGuestOSProfile [[-Name] <String> ] [-VMMServer <ServerConnection> ] [ <CommonParameters> ]
```

Parameter Set: SysprepScript

```
Get-SCGuestOSProfile [[-Name] <String> ] -AnswerFile <Script> [-OperatingSystem <OperatingSystem> ] [ <CommonParameters> ]
```

### Detailed Description

The Get-SCGuestOSProfile cmdlet gets one or more guest operating system profile objects from the VMM library.

### Parameters

#### **-AnswerFile<Script>**

Specifies a script object stored in the VMM library to use as an answer file. The name of the answer file script depends on the operating system that you want to install on a virtual machine:

ANSWER FILE    GUEST OS TO INSTALL ON VM

-----    -----

Sysprep.inf    Windows XP, Windows Server 2000, or Windows Server 2003

Unattend.xml    Windows Vista, Windows 7, or Windows Server 2008

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)

---

Accept Wildcard Characters?	false
-----------------------------	-------

### **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	false
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-OperatingSystem<OperatingSystem>**

Specifies the type of operating system for a virtual machine. To list the names of all available operating systems in VMM, type: "Get-SCOperatingSystem".

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none

---

Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **GuestOSProfile**

## Examples

### 1: Get all guest operating system profiles from the library.

This command gets all guest OS profile objects from the library on VMMServer01 and displays information about these profiles to the user.

```
PS C:\> Get-SCGuestOSProfile -VMMServer "VMMServer01.Contoso.com"
```

### 2: Get a specific guest operating system profile from the library.

This command gets the guest OS profile object named NewOSProfile01 and displays information about this profile to the user.

```
PS C:\> Get-GuestOSProfile -Name "NewOSProfile01"
```

## Related topics

[New-SCGuestOSProfile](#)

[Remove-SCGuestOSProfile](#)

[Set-SCGuestOSProfile](#)

---

# Get-SCHardwareProfile

---

## Get-SCHardwareProfile

Gets hardware profile objects from the VMM library.

### Syntax

Parameter Set: Connection

```
Get-SCHardwareProfile [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: All

```
Get-SCHardwareProfile [-All] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ID

```
Get-SCHardwareProfile [-ID <Guid> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCHardwareProfile cmdlet gets one or more hardware profile objects from the VMM library. You can use a hardware profile repeatedly to create new virtual machines or virtual machine templates.

### Parameters

#### -All

Retrieves a full list of all subordinate objects independent of the parent object. For example, the command Get-SCVirtualDiskDrive -All retrieves all virtual disk drive objects regardless of the virtual machine object or template object that each virtual disk drive object is associated with.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false



---

## **-ID<Guid>**

Specifies the numerical identifier (as a globally unique identifier, or GUID) for a specific object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **HardwareProfile**

---

## Examples

### 1: Get all hardware profiles from the library.

This command gets all hardware profile objects from the library on VMMServer01 and displays information about these profiles to the user.

```
PS C:\> Get-SCHardwareProfile -VMMServer "VMMServer01.Contoso.com"
```

### 2: Get a specific hardware profile from the library.

This command gets the hardware profile object named NewHWProfile01 and displays information about this hardware profile to the user.

```
C:\> Get-SCHardwareProfile | where { $_.Name -eq "NewHWProfile01" }
```

## Related topics

[New-SCHardwareProfile](#)

[Remove-SCHardwareProfile](#)

[Set-SCHardwareProfile](#)

---

# Get-SCHostReserve

---

## Get-SCHostReserve

Gets the host reserve and placement settings for a host group.

### Syntax

Parameter Set: FromHostGroup

```
Get-SCHostReserve -VMHostGroup <HostGroup> [-VMMServer <ServerConnection> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The Get-SCHostReserve cmdlet retrieves the placement settings for a host group. In addition to host reserve calculations, placement settings are used for the dynamic optimization and power optimization features.

### Parameters

#### **-VMHostGroup<HostGroup>**

Specifies a virtual machine host group object or an array of host group objects.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

#### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **-Confirm**

Prompts you for confirmation before executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- 
- **ConfigurationProvider**

## Examples

### 1: Get the Host reserve for a specified host group.

The first command gets the host group object named HostGroup01 and stores the object in the \$HostGroup variable.

The second command gets the host reserve object for the host group stored in \$HostGroup, and then stores the object in the \$HostReserve variable.

The last command displays information about the host reserve settings stored in \$HostReserve to the user.

```
PS C:\> $HostGroup = Get-SCVMHostGroup "HostGroup01"
```

```
PS C:\> $HostReserve = Get-SCHostReserve -VMHostGroup $HostGroup
```

```
PS C:\> $HostReserve
```

## Related topics

[Get-SCVMHostGroup](#)

[Set-SCHostReserve](#)

---

# Get-SCIPAddress

---

## Get-SCIPAddress

Gets allocated static and virtual IP addresses.

### Syntax

Parameter Set: All

```
Get-SCIPAddress [-All] [-Assigned] [-IsMulticast] [-UnAssigned] [-VMMServer  
<ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ByAllocatedToObjectID

```
Get-SCIPAddress -GrantToObjectID <Guid> [-Assigned] [-IsMulticast] [-UnAssigned] [-VMMServer  
<ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ByID

```
Get-SCIPAddress -ID <Guid> [-Assigned] [-IsMulticast] [-UnAssigned] [-VMMServer  
<ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ByIPAddress

```
Get-SCIPAddress -IPAddress <String> [-Assigned] [-IsMulticast] [-UnAssigned] [-VMMServer  
<ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ByPool

```
Get-SCIPAddress -StaticIPAddressPool <StaticIPAddressPool> [-Assigned] [-IsMulticast] [-  
UnAssigned] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

## Detailed Description

The Get-SCIPAddress cmdlet gets allocated static IP and virtual IP addresses.

## Parameters

### -All

Retrieves a full list of all subordinate objects independent of the parent object. For example, the command `Get-SCVirtualDiskDrive -All` retrieves all virtual disk drive objects regardless of the virtual machine object or template object that each virtual disk drive object is associated with.

Aliases	none
---------	------

---

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Assigned**

Retrieves IP addresses or MAC addresses that have been allocated from an address pool and assigned to a resource.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-GrantToObjectID<Guid>**

Specifies the ID of an object to which an allocated IP address or MAC address will be assigned.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-ID<Guid>**

Specifies the numerical identifier (as a globally unique identifier, or GUID) for a specific object.

---

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-IPAddress<String>**

Specifies an IPv4 or IPv6 address.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-IsMulticast**

Indicates that the IP address is a multicast address or that the IP address pool contains a multicast IP address range.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-StaticIPAddressPool<StaticIPAddressPool>**

Specifies an IP address pool from which you can assign static IP addresses.



Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-UnAssigned**

Retrieves IP addresses or MAC addresses that have been allocated from an address pool but not assigned to a resource.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **CloudDRPairingInfoData**

## Examples

### 1: Get all allocated IP addresses for a specific IP address pool.

The first command gets the IP address pool object with the subnet of 10.0.0.0/24 and stores it in the \$IPAddressPool variable.

The second command returns all allocated IP addresses for the IP address pool stored in \$IPAddressPool.

```
PS C:\> $IPAddressPool = Get-SCStaticIPAddressPool -IPv4 -Subnet "10.0.0.0/24"
```

```
PS C:\> Get-SCIPAddress -StaticIPAddressPool $IPAddressPool
```

## Related topics

[Grant-SCIPAddress](#)

[Revoke-SCIPAddress](#)

[Set-SCIPAddress](#)

---

# Get-SCISO

---

## Get-SCISO

Gets ISO files from the VMM library.

### Syntax

Parameter Set: All

```
Get-SCISO [-All] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: EquivalentResourceParamSet

```
Get-SCISO -FamilyName <String> [-Release <String> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ID

```
Get-SCISO [-ID <Guid> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: NameParamSet

```
Get-SCISO -Name <String> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCISO cmdlet gets ISO files from the VMM library. ISO files are stored in library shares on library servers.

In Virtual Machine Manager (VMM), some typical uses of an ISO file include:

- Storing an operating system ISO in the library that you can use later to install that operating system on a new or existing virtual machine deployed on a host.
- Storing application software, such as a Microsoft Office ISO, in the library, so that you can install it later on a virtual machine deployed on a host.

### Parameters

#### -All

Retrieves a full list of all subordinate objects independent of the parent object. For example, the command `Get-SCVirtualDiskDrive -All` retrieves all virtual disk drive objects regardless of the virtual machine object or template object that each virtual disk drive object is associated with.

---

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-FamilyName<String>**

Specifies a family name for a physical resource in the VMM library. This value is used in conjunction with Release, Namespace, and Type to establish equivalency among library resources.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-ID<Guid>**

Specifies the numerical identifier (as a globally unique identifier, or GUID) for a specific object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Release<String>**

Specifies a string that describes the release of a library resource. VMM automatically creates a release value for every resource imported into the library. After the resource has been imported, the string can be customized.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- ISO

## Examples

### 1: Get all ISOs on all VMM library servers.

This command gets all ISO objects on VMMServer01 and displays information about these ISOs to the user.

The ISO files that the retrieved objects represent are stored in library shares on library servers.

```
PS C:\> Get-SCISO -VMMServer "VMMServer01.Contoso.com"
```

### 2: Get all ISOs on a specific VMM library server.

This command gets all objects that represent ISO files stored on LibraryServer01 and displays information about these ISOs to the user.

```
PS C:\> Get-SCISO -VMMServer "VMMServer1.Contoso.com" | where { $_.LibraryServer.Name -eq "LibraryServer01.Contoso.com" }
```

### 3: Get all ISOs with a specific string in the file name on any VMM library server.

This command gets all ISO objects on any VMM library server managed by VMMServer01 that contain "OsISO" in the file name, and then displays information about these ISO objects to the user.

NOTE: By default, the name of an ISO object in the library is the same name as the name of the actual ISO file stored in the file system on the library server.

```
PS C:\> Get-SCISO -VMMServer "VMMServer01.Contoso.com" | where { $_.Name -match "OsISO" }
```

## Related topics

[Remove-SCISO](#)

[Set-SCISO](#)

---

# Get-SCISOConfiguration

---

## Get-SCISOConfiguration

Gets the ISO configuration for a virtual machine configuration.

### Syntax

Parameter Set: Default

```
Get-SCISOConfiguration -VMConfiguration <VMConfiguration> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Get-ISOConfiguration cmdlet gets the ISO configuration for a virtual machine configuration. An ISO configuration allows you to customize the source and target locations for the physical resources that are needed to create a virtual machine.

### Parameters

#### **-VMConfiguration<VMConfiguration>**

Specifies a virtual machine configuration object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

#### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

---

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **ISOConfiguration**

## Examples

### 1. Get the ISO configuration for a given virtual machine configuration.

The first command gets the service configuration object named Service01 and stores the object in the \$ServiceConfig variable.

The second command gets the computer tier configuration object for the the service configuration in \$ServiceConfig and stores the object in the \$TierConfig variable.

The third command gets the virtual machine configuration for the computer tier configuration stored in \$TierConfig and stores the object in the \$VMConfig variable.

The fourth command gets the ISO configuration for the first virtual machine configuration stored in \$VMConfig and stores the object in the \$ISOConfig variable.

The last command displays the properties of the ISO configuration stored in \$ISOConfig to the user.

```
PS C:\> $ServiceConfig = Get-SCServiceConfiguration -Name "Service01"
```

```
PS C:\> $TierConfig = Get-SCComputerTierConfiguration -ServiceConfiguration $ServiceConfig
```

```
PS C:\> $VMConfig = Get-SCVMConfiguration -ComputerTierConfiguration $TierConfig
```

```
PS C:\> $ISOConfig = Get-SCISOConfiguration -VMConfiguration $VMConfig
```

```
PS C:\> $ISOConfig
```



---

## Related topics

[Set-SCISOConfiguration](#)

---

# Get-SCJob

---

## Get-SCJob

Gets VMM job objects.

### Syntax

Parameter Set: NewestJobs

```
Get-SCJob [[-Name] <String> ] [-Full] [-ID <Guid> ] [-Job <Task> ] [-Newest <Int32> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: AllJobs

```
Get-SCJob [[-Name] <String> ] -All [-Full] [-ID <Guid> ] [-Job <Task> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: RunningJobs

```
Get-SCJob [[-Name] <String> ] -Running [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCJob cmdlet gets one or more Virtual Machine Manager (VMM) job objects on the VMM server. A job is a series of steps that are performed sequentially to complete an action in the VMM environment. You can retrieve job objects based on specified criteria.

In VMM, you can group a series of jobs and run them together as a set. For example, a complex action in VMM, such as creating a template, might incorporate a series of jobs, known as a job group. For examples that show you how to use job groups, see the following cmdlets: New-SCVMTemplate, New-SCHardwareProfile, New-SCVirtualDiskDrive, New-SCVirtualDVDDrive, New-SCVirtualMachine, and Set-SCVirtualCOMPort.

### Parameters

#### -All

Retrieves a full list of all subordinate objects independent of the parent object. For example, the command Get-SCVirtualDiskDrive -All retrieves all virtual disk drive objects regardless of the virtual machine object or template object that each virtual disk drive object is associated with.

Aliases	none
---------	------

---

Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Full**

Specifies that the cmdlet returns the job object with an audit record.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-ID<Guid>**

Specifies the numerical identifier (as a globally unique identifier, or GUID) for a specific object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Job<Task>**

Specifies a VMM job object.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	false
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Newest<Int32>**

Returns all jobs created in the last specified number of hours, or returns the specified number of most recent software updates.

Example format to return all jobs created in the last 48 hours: `Get-SCJob -Newest 48`

Example format to return the 10 newest updates: `Get-SCUpdate -Newest 10`

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Running**

Indicates that all running jobs are returned.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **Job**

## **Examples**

### **1: Get all running jobs.**

This command retrieves all running jobs.

```
PS C:\> PS C:> Get-Job -Running
```

---

## 2: Get all failed jobs.

The first command gets all job objects with a status of Failed, and then uses the pipeline operator to pass each object to the Format-List cmdlet which stores the name, ID, and Status in the \$VMMJobs variable.

The second command displays the information stored in \$VMMJobs to the user (in this case, that is the name, ID and Status of each running job).

```
PS C:\> $VMMJobs = Get-SCJob | where { $_.Status -eq "Failed" } | Format-List -Property  
Name, ID, Status  
PS C:\> $VMMJobs
```

## 3: Get information about the .NET type, methods, and properties of VMM job objects.

This command uses the Get-Member cmdlet to display the .NET type, properties, methods, and events for Get-SCJob.

```
PS C:\> PS C:> Get-SCJob | Get-Member
```

## Related topics

[Find-SCJob](#)

[Get-SCStep](#)

[Restart-SCJob](#)

[Stop-SCJob](#)

---

# Get-SCLibraryRating

---

## Get-SCLibraryRating

Calculates the placement rating of library servers to determine whether a SAN transfer can be used to transfer a virtual machine from a host to the library.

### Syntax

Parameter Set: FromLibraryServers

```
Get-SCLibraryRating -LibraryServer <LibraryServer[]> [-VM <VM> ] [-VMMServer  
<ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCLibraryRating cmdlet calculates the placement rating of library servers managed by Virtual Machine Manager (VMM). Specifically, this rating indicates whether VMM can use SAN transfer to transfer a particular virtual machine from a host server to a library server. If a SAN transfer is not possible, you can use a LAN transfer to store the virtual machine in the library.

For information about how to store a virtual machine in the VMM library, type "Get-Help Save-SCVirtualMachine -detailed".

### Parameters

#### **-LibraryServer<LibraryServer[]>**

Specifies a VMM library server object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

---

## **-VM<VM>**

Specifies a virtual machine object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## **Examples**

### **1: Determine whether you can use a SAN transfer to store a virtual machine on the specified library server.**

The first command gets the virtual machine object named VM01 and stores the object in the \$VM variable.

The second command gets the library server object named LibraryServer01 and stores the object in the \$LibServer variable.



---

The third command gets the placement rating for LibraryServer01 (which indicates whether VMM can use a SAN transfer to transfer VM01 to LibraryServer01) and stores the rating object in the \$LibRating variable.

The last command displays the rating stored in \$LibRating to the user.

TIP: If a SAN transfer is not possible, you can use a LAN transfer to store the virtual machine in the library.

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM01"
PS C:\> $LibServer = Get-SCLibraryServer -ComputerName "LibraryServer01.Contoso.com"
PS C:\> $LibRating = Get-SCLibraryRating -LibraryServer $LibServer -VM $VM
PS C:\> $LibRating
```

## 2: Get placement ratings for each available library server.

The first command gets the virtual machine object named VM01 and stores the object in the \$VM variable.

The second command gets all library server objects that are managed by VMM and stores the objects in the \$LibServer array.

The fourth command returns the placement rating for each library server object in \$LibServer (which indicates whether VMM can use a SAN transfer to transfer VM01 to each of the library servers) and stores the rating for each library server object in the \$LibRating array.

The last command displays the rating information for each library object to the user.

TIP: If a SAN transfer is not possible, you can use a LAN transfer to store the virtual machine in the library.

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM01"
PS C:\> $LibServer = Get-SCLibraryServer
PS C:\> $LibRating = Get-SCLibraryRating -LibraryServer $LibServer -VM $VM
PS C:\> $LibRating
```

## Related topics

[Get-SCLibraryServer](#)

[Get-SCVirtualMachine](#)

[Save-SCVirtualMachine](#)

---

# Get-SCLibraryServer

---

## Get-SCLibraryServer

Gets VMM library server objects.

### Syntax

Parameter Set: Default

```
Get-SCLibraryServer [[-ComputerName] <String> ] [-ID <Guid> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCLibraryServer cmdlet gets one or more library server objects from the VMM library.

For more information about library servers, type: "Get-Help Add-LibraryServer -detailed".

### Parameters

#### **-ComputerName<String>**

Specifies the name of a computer that VMM can uniquely identify on your network. Valid formats are: FQDN, IPv4 or IPv6 address, or NetBIOS name.

NOTE: See the examples for a specific cmdlet to determine how that cmdlet specifies the computer name.

Aliases	none
Required?	false
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

#### **-ID<Guid>**

Specifies the numerical identifier (as a globally unique identifier, or GUID) for a specific object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **LibraryServer**

## **Examples**

### **1: Get all library servers.**

This command gets all library server objects on VMMServer01 and displays information about these library servers to the user.

NOTE: The name of a library server is the same as its computer name.

```
PS C:\> Get-SCLibraryServer -VMMServer "VMMServer01.Contoso.com"
```

---

## 2: Get a specific library server.

This command gets the library object named LibraryServer01 on VMMServer01 and displays information about this library server to the user.

```
PS C:\> Get-SCLibraryServer -VMMServer "VMMServer01.Contoso.com" -ComputerName  
"LibraryServer01.Contoso.com"
```

## 3: Get all library servers that match specified criteria.

This command gets all library server objects on VMMServer01 whose name includes the string "LibraryServer" (such as LibraryServer01 and LlibraryServer02) and stores these library server objects in the \$LibServers variable.

```
PS C:\> $LibServers = Get-SCLibraryServer -VMMServer "VMMServer01.Contoso.com" | where {  
$_.Name -match "LibraryServer" }
```

## Related topics

[Add-SCLibraryServer](#)

[Add-SCLibraryShare](#)

[Remove-SCLibraryServer](#)

[Set-SCLibraryServer](#)

---

# Get-SCLibraryShare

---

## Get-SCLibraryShare

Gets VMM library shares.

### Syntax

Parameter Set: Default

```
Get-SCLibraryShare [-ID <Guid> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCLibraryShare cmdlet gets VMM library shares.

A library share is a Windows share on a VMM library server that is used to store files that contain library resources. Resources can include virtual machine templates, hardware profiles, guest operating system profiles, virtual hard disks (Windows-based .vhd files, Citrix XenServer-based .vhd files or VMware-based .vmdk files), virtual floppy disks (Windows-based .vfd files or VMware-based .flp files), ISO images (.iso files), and scripts, as well as stored virtual machines.

For more information about library shares, type: "Get-Help Add-LibraryShare -detailed".

### Parameters

#### **-ID<Guid>**

Specifies the numerical identifier (as a globally unique identifier, or GUID) for a specific object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **LibraryShare**

## **Examples**

### **1: Get all library shares.**

This command gets all library share objects from the VMM library on VMMServer01 and displays information about these library shares to the user.

```
PS C:\> Get-SCLibraryShare -VMMServer "VMMServer01.Contoso.com"
```

### **2: Get a specific library share on the specified library server.**

This command gets the library share object named AllVHDs (on library server LibraryServer01) from the library on VMMServer01 and then stores the share object in the \$LibShare variable.

```
PS C:\> $LibShare = Get-SCLibraryShare -VMMServer "VMMServer01.Contoso.com" | where {  
$_.LibraryServer.Name -eq "LibraryServer01.Contoso.com" -and $_.Name -eq "AllVHDs" }
```

---

### 3: Get all library shares on a specific library server.

The first command retrieves the library server object named LibraryServer01 from the library on VMMServer01 and stores it in the \$LibServer variable.

The second command gets all library share objects on LibraryServer01 and stores the objects in the \$AllLibShares variable.

The last command passes each object in \$AllLibShares to the Get-Member cmdlet, which displays the .NET type for a library share object and the list of methods and properties that are associated with a VMM library share object.

```
PS C:\> $LibServer = Get-SCLibraryServer -VMMServer "VMMServer01.Contoso.com" -ComputerName "LibraryServer01.Contoso.com"
```

```
PS C:\> $AllLibShares = Get-SCLibraryShare | where { $_.LibraryServer.Name -eq "$LibServer" }
```

```
PS C:\> $AllLibShares | Get-Member
```

### Related topics

[Add-SCLibraryShare](#)

[Find-SCLibraryShare](#)

[Get-SCLibraryServer](#)

[Read-SCLibraryShare](#)

[Remove-SCLibraryShare](#)

[Set-SCLibraryShare](#)

---

# Get-SCLoadBalancer

---

## Get-SCLoadBalancer

Gets a load balancer object.

### Syntax

Parameter Set: GlobalList

```
Get-SCLoadBalancer [[-LoadBalancerAddress] <String> ] [-All] [-LogicalNetwork <LogicalNetwork[]> ] [-Manufacturer <String> ] [-Model <String> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: AccessibleToCloudRootHostGroup

```
Get-SCLoadBalancer [[-LoadBalancerAddress] <String> ] -CloudRootHostGroup <HostGroup[]> [-LogicalNetwork <LogicalNetwork[]> ] [-Manufacturer <String> ] [-Model <String> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: AccessibleToCloudRootVMwareResourcePool

```
Get-SCLoadBalancer [[-LoadBalancerAddress] <String> ] -CloudRootVMwareResourcePool <VmwareResourcePool> [-LogicalNetwork <LogicalNetwork[]> ] [-Manufacturer <String> ] [-Model <String> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ByCloud

```
Get-SCLoadBalancer [[-LoadBalancerAddress] <String> ] -Cloud <Cloud> [-LogicalNetwork <LogicalNetwork[]> ] [-Manufacturer <String> ] [-Model <String> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ByHostGroup

```
Get-SCLoadBalancer [[-LoadBalancerAddress] <String> ] -VMHostGroup <HostGroup> [-LogicalNetwork <LogicalNetwork[]> ] [-Manufacturer <String> ] [-Model <String> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ByID

```
Get-SCLoadBalancer [[-LoadBalancerAddress] <String> ] -ID <Guid> [-LogicalNetwork <LogicalNetwork[]> ] [-Manufacturer <String> ] [-Model <String> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCLoadBalancer cmdlet gets one or more load balancer objects.



---

## Parameters

### **-All**

Retrieves a full list of all subordinate objects independent of the parent object. For example, the command `Get-SCVirtualDiskDrive -All` retrieves all virtual disk drive objects regardless of the virtual machine object or template object that each virtual disk drive object is associated with.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Cloud<Cloud>**

Specifies a private cloud object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-CloudRootHostGroup<HostGroup[]>**

Specifies a host group that is defined at the root level for a private cloud.

Aliases	none
Required?	true
Position?	named
Default Value	none

---

Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-CloudRootVMwareResourcePool<VmwareResourcePool>**

Specifies a VMware resource pool that is defined at the root level for a private cloud.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-ID<Guid>**

Specifies the numerical identifier (as a globally unique identifier, or GUID) for a specific object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-LoadBalancerAddress<String>**

Specifies the fully qualified domain name (FQDN) or IP address of a load balancer. Usual formats are FQDN, IPv4 or IPv6 addresses, but check with the load balancer manufacturer for the valid format for your load balancer.

Aliases	none
Required?	false

Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-LogicalNetwork<LogicalNetwork[]>**

Specifies a logical network. A logical network is a named grouping of IP subnets and VLANs that is used to organize and simplify network assignments.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Manufacturer<String>**

Specifies the name of the company that manufactured a physical device.

Valid characters include: letters (a-z), numbers (0-9), underscore (\_), hyphen(-), dot(.), and single quote(').

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Model<String>**

Specifies the model of a physical device.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMHostGroup<HostGroup>**

Specifies a virtual machine host group object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

---

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **LoadBalancer**

## Examples

### 1: Get all load balancers for a host group.

The first command gets the host group object named HostGroup01 and stores the object in the \$HostGroup variable.

The second command gets all load balancer objects accessible to the host group stored in \$HostGroup and stores the objects in the \$LoadBalancers variable.

The last command displays information about each of the load balancers stored in \$LoadBalancers to the user.

```
PS C:\> $HostGroup = Get-SCVMHostGroup | where { $_.Path -eq "All Hosts\HostGroup01" }
```

```
PS C:\> $LoadBalancers = Get-SCLoadBalancer -VMHostGroup $HostGroup
```

```
PS C:\> $LoadBalancers
```

### 2: Get all load balancers of a given type for a host group.

The first command gets the host group object named Production and stores the object in the \$HostGroup variable.

The second command gets the load balancer objects with the specified manufacturer and model accessible to the host group stored in \$HostGroup and stores the objects in the \$LoadBalancers variable.

The last command displays information about each load balancer object stored in \$LoadBalancers to the user.

```
PS C:\> $HostGroup = Get-SCVMHostGroup | where { $_.Path -eq "All Hosts\HostGroup02\Production" }
```

```
PS C:\> $LoadBalancers = Get-SCLoadBalancer -VMHostGroup $HostGroup -Manufacturer "LB Manufacturer" -Model "LB01"
```

```
PS C:\> $LoadBalancers
```

## Related topics

[Add-SCLoadBalancer](#)

[Read-SCLoadBalancer](#)

[Remove-SCLoadBalancer](#)

[Set-SCLoadBalancer](#)

[Test-SCLoadBalancer](#)

---

# Get-SCLoadBalancerConfiguration

---

## Get-SCLoadBalancerConfiguration

Gets the configuration details for the load balancer that is contained within a computer tier configuration.

### Syntax

Parameter Set: Default

```
Get-SCLoadBalancerConfiguration -ComputerTierConfiguration <ComputerTierConfiguration> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCLoadBalancerConfiguration cmdlet gets the configuration details for the load balancer that is contained within a computer tier configuration.

### Parameters

#### **-ComputerTierConfiguration<ComputerTierConfiguration>**

Specifies a computer tier configuration object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

#### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

---

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **LoadBalancerConfiguration**

## Examples

### 1: Get the load balancer configuration for a computer tier configuration.

The first command gets service the service configuration object named Service01 and stores the object in the \$ServiceConfig variable.

The second command gets the computer tier configuration for the service configuration stored in \$ServiceConfig and stores the object in the \$TierConfig variable.

The third command gets the load balancer configuration for the computer tier configuration stored in \$TierConfig and stores the object in the \$LBConfig variable.

The last command displays the properties of the load balancer configuration stored in \$LBConfig to the user.

```
PS C:\> $ServiceConfig = Get-SCServiceConfiguration -Name "Service01"
```

```
PS C:\> $TierConfig = Get-SCComputerTierConfiguration -ServiceConfiguration $ServiceConfig
```

```
PS C:\> $LBConfig = Get-SCLoadBalancerConfiguration -ComputerTierConfiguration $TierConfig
```

```
PS C:\> $LBConfig
```

## Related topics

[Set-SCLoadBalancerConfiguration](#)

---

# Get-SCLoadBalancerTemplate

---

## Get-SCLoadBalancerTemplate

Gets a load balancer template for a service or computer tier template.

### Syntax

Parameter Set: All

```
Get-SCLoadBalancerTemplate [-All] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ComputerTierTemplate

```
Get-SCLoadBalancerTemplate -ComputerTierTemplate <ComputerTierTemplate> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ID

```
Get-SCLoadBalancerTemplate [-ID <Guid> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ServiceTemplate

```
Get-SCLoadBalancerTemplate -ServiceTemplate <ServiceTemplate> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCLoadBalancerTemplate cmdlet gets the load balancer template for a service template or a computer tier template.

### Parameters

#### -All

Retrieves a full list of all subordinate objects independent of the parent object. For example, the command Get-SCVirtualDiskDrive -All retrieves all virtual disk drive objects regardless of the virtual machine object or template object that each virtual disk drive object is associated with.

Aliases	none
Required?	false
Position?	named



---

Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-ComputerTierTemplate<ComputerTierTemplate>**

Specifies a computer tier template object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-ID<Guid>**

Specifies the numerical identifier (as a globally unique identifier, or GUID) for a specific object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-ServiceTemplate<ServiceTemplate>**

Specifies a service template object.

Aliases	none
Required?	true
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **LoadBalancerTemplate**

## **Examples**

### **1: Get the load balancer template for a service template.**

The first command gets the service template object named ServiceTemplate01 and stores the object in the \$ServiceTemplate variable.

The second command gets the load balancer template for the service template stored in \$ServiceTemplate and stores the object in the \$LoadBalancerTemplate variable.

The last command displays the properties of the load balancer template stored in \$LoadBalancerTemplate to the user.

```
PS C:\> $ServiceTemplate = Get-SCServiceTemplate -Name "ServiceTemplate01"
```

```
PS C:\> $LoadBalancerTemplate = Get-SCLoadBalancerTemplate -ServiceTemplate $ServiceTemplate
```

```
PS C:\> $LoadBalancerTemplate
```

---

## 2: Get the load balancer template for a computer tier template.

The first command gets the service template object named ServiceTemplate01 and stores the object in the \$ServiceTemplate variable.

The second command gets the computer tier template for the service template stored in \$ServiceTemplate and stores the object in the \$TierTemplate variable.

The third command gets the load balancer template for the computer tier template stored in \$TierTemplate and stores the object in the \$LoadBalancerTemplate variable.

The last command displays the properties of the load balancer template stored in \$LoadBalancerTemplate to the user.

```
PS C:\> $ServiceTemplate = Get-SCServiceTemplate -Name "ServiceTemplate01"
```

```
PS C:\> $TierTemplate = Get-SCComputerTierTemplate -ServiceTemplate $ServiceTemplate
```

```
PS C:\> $LoadBalancerTemplate = Get-SCLoadBalancerTemplate -ComputerTierTemplate  
$TierTemplate
```

```
PS C:\> $LoadBalancerTemplate
```

### Related topics

[Get-SCComputerTierTemplate](#)

[Get-SCServiceTemplate](#)

[New-SCLoadBalancerTemplate](#)

[Remove-SCLoadBalancerTemplate](#)

[Set-SCLoadBalancerTemplate](#)

---

# Get-SCLoadBalancerVIP

---

## Get-SCLoadBalancerVIP

Gets a load balancer VIP object.

### Syntax

Parameter Set: GlobalList

```
Get-SCLoadBalancerVIP [[-Name] <String> ] [-IPAddress <String> ] [-LoadBalancer <LoadBalancer> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ByID

```
Get-SCLoadBalancerVIP -ID <Guid> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCLoadBalancerVIP cmdlet gets one or more load balancer virtual IP (VIP) objects.

### Parameters

#### **-ID<Guid>**

Specifies the numerical identifier (as a globally unique identifier, or GUID) for a specific object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

#### **-IPAddress<String>**

Specifies an IPv4 or IPv6 address.

---

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-LoadBalancer<LoadBalancer>**

Specifies a load balancer object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	false
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

---

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **LoadBalancerVIP**

## Examples

### 1: Get a load balancer virtual IP (VIP) for a specific load balancer.

The first command gets the load balancer object with the address LB01.Contoso.com and stores the object in the \$LoadBalancer variable.

The second command gets the load balancer VIP for the load balancer stored in \$LoadBalancer with an IP address of 192.168.0.1.

```
PS C:\> $LoadBalancer = Get-SCLoadBalancer -LoadBalancerAddress "LB01.Contoso.com"
```

```
PS C:\> Get-SCLoadBalancerVIP -LoadBalancer $LoadBalancer -IPAddress "192.168.0.1"
```

## Related topics

[New-SCLoadBalancerVIP](#)

[Read-SCLoadBalancerVIP](#)

[Remove-SCLoadBalancerVIP](#)

---

# Get-SCLoadBalancerVIPMember

---

## Get-SCLoadBalancerVIPMember

Gets a member of a load balancer VIP.

### Syntax

Parameter Set: All

```
Get-SCLoadBalancerVIPMember [-IPAddress <String> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ByID

```
Get-SCLoadBalancerVIPMember -ID <Guid> [-IPAddress <String> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ByLoadBalancerVIP

```
Get-SCLoadBalancerVIPMember -LoadBalancerVIP <LoadBalancerVIP> [-IPAddress <String> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

## Detailed Description

The Get-SCLoadBalancerVIPMember cmdlet gets a member of a load balancer virtual IP (VIP).

## Parameters

### -ID<Guid>

Specifies the numerical identifier (as a globally unique identifier, or GUID) for a specific object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-IPAddress<String>**

Specifies an IPv4 or IPv6 address.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-LoadBalancerVIP<LoadBalancerVIP>**

Specifies a virtual IP (VIP) in a load balancer.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false



---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **LoadBalancerVIPMember**

## Examples

### 1: Get a load balancer virtual IP (VIP) member.

The first command gets the load balancer object with the address LB01.Contoso.com and stores the object in the \$LoadBalancer variable.

The second command gets the load balancer VIP with the IP address of 192.168.0.1 for the load balancer stored in \$LoadBalancer and stores the object in the \$VIP variable.

The last command gets the load balancer member for the load balancer VIP stored in \$VIP with the IP address of 192.168.0.1 and displays information about the member to the user.

```
PS C:\> $LoadBalancer = Get-SCLoadBalancer -LoadBalancerAddress "LB01.Contoso.com"
```

```
PS C:\> $VIP = Get-SCLoadBalancerVIP -LoadBalancer $LoadBalancer -IPAddress "192.168.0.1"
```

```
PS C:\> Get-SCLoadBalancerVIPMember -LoadBalancerVIP $VIP -IPAddress "192.168.0.1"
```

## Related topics

[Disable-SCLoadBalancerVIPMember](#)

[Enable-SCLoadBalancerVIPMember](#)

[Get-SCLoadBalancerVIP](#)

[New-SCLoadBalancerVIPMember](#)

[Remove-SCLoadBalancerVIPMember](#)

---

# Get-SCLoadBalancerVIPTemplate

---

## Get-SCLoadBalancerVIPTemplate

Gets a load balancer VIP template.

### Syntax

Parameter Set: Global

```
Get-SCLoadBalancerVIPTemplate [-Manufacturer <String> ] [-Model <String> ] [-Name <String> ]  
[-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ByCloud

```
Get-SCLoadBalancerVIPTemplate -Cloud <Cloud> [-VMMServer <ServerConnection> ] [  
<CommonParameters>]
```

Parameter Set: ByID

```
Get-SCLoadBalancerVIPTemplate -ID <Guid> [-VMMServer <ServerConnection> ] [  
<CommonParameters>]
```

### Detailed Description

The Get-SCLoadBalancerVIPTemplate gets one or more load balancer virtual IP (VIP) templates.

### Parameters

#### **-Cloud<Cloud>**

Specifies a private cloud object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-ID<Guid>**

Specifies the numerical identifier (as a globally unique identifier, or GUID) for a specific object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Manufacturer<String>**

Specifies the name of the company that manufactured a physical device.

Valid characters include: letters (a-z), numbers (0-9), underscore (\_), hyphen(-), dot(.), and single quote(').

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Model<String>**

Specifies the model of a physical device.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **LoadBalancerVIPTemplate**

---

## Examples

### 1: Retrieve virtual IP (VIP) templates associated with a specific load balancer manufacturer and model.

This command gets all virtual IP templates associated with the manufacturer LB Manufacturer and the model LB01, and then displays information about these virtual IP templates to the user.

```
PS C:\> Get-SCLoadBalancerVIPTemplate -Manufacturer "LB Manufacturer" -Model "LB01"
```

## Related topics

[New-SCLoadBalancerVIPTemplate](#)

[Remove-SCLoadBalancerVIPTemplate](#)

[Set-SCLoadBalancerVIPTemplate](#)

---

# Get-SCLogicalNetwork

---

## Get-SCLogicalNetwork

Gets a logical network object.

### Syntax

Parameter Set: GlobalList

```
Get-SCLogicalNetwork [[-Name] <String> ] [-All] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: AccessibleToCloudRootHostGroup

```
Get-SCLogicalNetwork [[-Name] <String> ] -CloudRootHostGroup <HostGroup[]> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: AccessibleToCloudRootVMwareResourcePool

```
Get-SCLogicalNetwork [[-Name] <String> ] -CloudRootVMwareResourcePool <VmwareResourcePool> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ByCloud

```
Get-SCLogicalNetwork [[-Name] <String> ] -Cloud <Cloud> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ByHost

```
Get-SCLogicalNetwork [[-Name] <String> ] -VMHost <Host> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ByHostGroup

```
Get-SCLogicalNetwork [[-Name] <String> ] -VMHostGroup <HostGroup> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ByID

```
Get-SCLogicalNetwork [[-Name] <String> ] [-ID <Guid> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCLogicalNetwork cmdlet retrieves one or more Virtual Machine Manager (VMM) logical network objects.

---

## Parameters

### **-All**

Retrieves a full list of all subordinate objects independent of the parent object. For example, the command `Get-SCVirtualDiskDrive -All` retrieves all virtual disk drive objects regardless of the virtual machine object or template object that each virtual disk drive object is associated with.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Cloud<Cloud>**

Specifies a private cloud object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-CloudRootHostGroup<HostGroup[]>**

Specifies a host group that is defined at the root level for a private cloud.

Aliases	none
Required?	true
Position?	named
Default Value	none

Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-CloudRootVMwareResourcePool<VmwareResourcePool>**

Specifies a VMware resource pool that is defined at the root level for a private cloud.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-ID<Guid>**

Specifies the numerical identifier (as a globally unique identifier, or GUID) for a specific object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	false
Position?	1
Default Value	none



---

Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMHost<Host>**

Specifies an array of virtual machine host objects.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMHostGroup<HostGroup>**

Specifies a virtual machine host group object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none

---

Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **LogicalNetwork**

## Examples

### 1: Retrieve all available logical networks for a specified host group.

The first command gets the host group object at the path "All Hosts\HostGroup01" and stores the object in the \$Hostgroup variable.

The second command gets all of the logical networks for the host group stored in \$Hostgroup and displays information about each logical network to the user.

```
PS C:\> $Hostgroup = Get-SCVMHostGroup | where { $_.Path -eq "All Hosts\HostGroup01" }
```

```
PS C:\> Get-SCLogicalNetwork -VMHostGroup $Hostgroup
```

## Related topics

[New-SCLogicalNetwork](#)

[Remove-SCLogicalNetwork](#)

[Set-SCLogicalNetwork](#)

---

# Get-SCLogicalNetworkDefinition

---

## Get-SCLogicalNetworkDefinition

Gets a logical network definition.

### Syntax

Parameter Set: All

```
Get-SCLogicalNetworkDefinition [[-Name] <String> ] [-LogicalNetwork <LogicalNetwork> ] [-ReturnAllTypes] [-Subnet <String> ] [-VLanID <Int32> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ByCloud

```
Get-SCLogicalNetworkDefinition [[-Name] <String> ] -Cloud <Cloud> [-LogicalNetwork <LogicalNetwork> ] [-ReturnAllTypes] [-Subnet <String> ] [-VLanID <Int32> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ByHostGroup

```
Get-SCLogicalNetworkDefinition [[-Name] <String> ] -VMHostGroup <HostGroup> [-LogicalNetwork <LogicalNetwork> ] [-ReturnAllTypes] [-Subnet <String> ] [-VLanID <Int32> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCLogicalNetworkDefinition cmdlet gets one or more logical network definitions. A logical network definition (also called a network site) can be associated with one or more logical networks.

### Parameters

#### **-Cloud<Cloud>**

Specifies a private cloud object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false

---

Accept Wildcard Characters?	false
-----------------------------	-------

### **-LogicalNetwork<LogicalNetwork>**

Specifies a logical network. A logical network is a named grouping of IP subnets and VLANs that is used to organize and simplify network assignments.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	false
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-ReturnAllTypes**

Returns logical network definitions of any isolation type.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Subnet<String>**

Specifies an IP subnet (IPv4 or IPv6) in Classless Inter-Domain Routing (CIDR) notation.

Example format for an IPv4 subnet: 192.168.0.1/24

Example format for an IPv6 subnet: FD4A:29CD:184F:3A2C::/64

NOTE: An IP subnet cannot overlap with any other subnet in a host group or child host groups.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VlanID<Int32>**

Assigns a numerical identifier in the range 1-4094 to a virtual network adapter on a virtual machine or to a physical network adapter on a virtual machine host.

Configure a VlanID on a Hyper-V, VMware ESX, or Citrix XenServer host:

- On an externally bound physical network adapter when the Vlan mode is Access.

Configure a VlanID on a virtual network adapter of a virtual machine:

- Bound to a physical network adapter on the host, or
- Bound to an internal virtual network on the host.

Example format: -VlanEnabled

-VlanMode "Access" -VLANID 35

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false

---

Accept Wildcard Characters?	false
-----------------------------	-------

### **-VMHostGroup<HostGroup>**

Specifies a virtual machine host group object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

### **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **LogicalNetworkDefiniton**

---

## Examples

### 1: Retrieve the logical network definition for a logical network.

The first command gets the logical network named "LogicalNetwork01" and stores it in the \$LogicalNetwork variable.

The second command gets the host group named "HostGroup01" and stores it in the \$HostGroup variable.

The third command gets the logical network definition for the logical network stored in \$LogicalNetwork and the host group stored in the \$HostGroup variable (including its parent host group if inheritance is enabled).

```
PS C:\> $LogicalNetwork = Get-SCLogicalNetwork -Name "LogicalNetwork01"
```

```
PS C:\> $HostGroup = Get-SCVMHostGroup | where { $_.Path -eq "All Hosts\HostGroup01" }
```

```
PS C:\> Get-SCLogicalNetworkDefinition -LogicalNetwork $LogicalNetwork -VMHostGroup $HostGroup
```

## Related topics

[New-SCLogicalNetworkDefinition](#)

[Remove-SCLogicalNetworkDefinition](#)

[Set-SCLogicalNetworkDefinition](#)

---

# Get-SCLogicalSwitch

---

## Get-SCLogicalSwitch

Gets a logical switch.

### Syntax

Parameter Set: All

```
Get-SCLogicalSwitch [[-Name] <String> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ById

```
Get-SCLogicalSwitch [[-Name] <String> ] [-ID <Guid> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCLogicalSwitch cmdlet gets a logical switch object.

### Parameters

#### **-ID<Guid>**

Specifies the numerical identifier (as a globally unique identifier, or GUID) for a specific object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

#### **-Name<String>**

Specifies the name of a VMM object.



Aliases	none
Required?	false
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## **Examples**

### **1: Get a logical switch by its name.**

This command gets the logical switch named LogicalSwitch01 and stores it in the \$LogicalSwitch variable.

```
PS C:\> $LogicalSwitch = Get-SCLogicalSwitch -Name "LogicalSwitch01"
```

## **Related topics**

[New-SCLogicalSwitch](#)

[Remove-SCLogicalSwitch](#)

---

[Set-SCLogicalSwitch](#)

---

# Get-SCMACAddress

---

## Get-SCMACAddress

Gets allocated MAC addresses.

### Syntax

Parameter Set: All

```
Get-SCMACAddress [-All] [-Assigned] [-UnAssigned] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ByAllocatedToObjectID

```
Get-SCMACAddress -GrantToObjectID <Guid> [-Assigned] [-UnAssigned] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ByID

```
Get-SCMACAddress -ID <Guid> [-Assigned] [-UnAssigned] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ByMACAddress

```
Get-SCMACAddress -MACAddress <String> [-Assigned] [-UnAssigned] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ByPool

```
Get-SCMACAddress -MACAddressPool <MACAddressPool> [-Assigned] [-UnAssigned] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

## Detailed Description

The Get-SCMACAddress cmdlet gets allocated MAC addresses.

## Parameters

### -All

Retrieves a full list of all subordinate objects independent of the parent object. For example, the command `Get-SCVirtualDiskDrive -All` retrieves all virtual disk drive objects regardless of the virtual machine object or template object that each virtual disk drive object is associated with.

Aliases	none
---------	------

---

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Assigned**

Retrieves IP addresses or MAC addresses that have been allocated from an address pool and assigned to a resource.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-GrantToObjectID<Guid>**

Specifies the ID of an object to which an allocated IP address or MAC address will be assigned.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-ID<Guid>**

Specifies the numerical identifier (as a globally unique identifier, or GUID) for a specific object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-MACAddress<String>**

Specifies the Media Access Control (MAC) address, or a set of MAC addresses, for a physical or virtual network adapter on a computer. Valid values are: one or more MAC addresses.

Example format for a single MAC address:

-MACAddress "00-15-5D-B4-DC-00"

Example format for a set of MAC addresses:

-MACAddress "00-15-5D-B4-DC-00", "00-1A-A0-E3-75-29"

Example format for a set of MAC addresses:

\$Macs = "00-15-5D-B4-DC-00", "00-1A-A0-E3-75-29"

Set-SCPXEServer -MACAddress \$Macs

NOTE: When used with New-SCPXEServer or Set-SCPXEServer, the MACAddress parameter updates the PXE interfaces from which the SCDM PXE Server listens for and responds to PXE requests

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-MACAddressPool<MACAddressPool>**

Specifies a MAC address pool.

Aliases	none
Required?	true

Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-UnAssigned**

Retrieves IP addresses or MAC addresses that have been allocated from an address pool but not assigned to a resource.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

---

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **CloudPairing[]**

## Examples

### 1: Get the allocated MAC addresses for a specific MAC address pool.

The first command gets the host group object at the path "All Hosts\HostGroup02\Production" and stores the object in the \$HostGroup variable.

The second command disables inheritance of network settings for the host group stored in \$HostGroup. This action returns only the MAC address pools associated with All Hosts\HostGroup02\Production in the next command. Otherwise, all MAC address pools inherited by this host group are also returned.

The third command gets the MAC address pool objects associated with the host group stored in \$HostGroup and stores the objects in the \$MACAddressPool array.

The last command gets the allocated MAC addresses for the first MAC address pool in \$MACAddressPool.

```
PS C:\> $HostGroup = Get-SCVMHostGroup | where { $_.Path -eq "All Hosts\HostGroup02\Production" }
```

```
PS C:\> Set-SCVMHostGroup -VMHostGroup $HostGroup -InheritNetworkSettings $False
```

```
PS C:\> $MACAddressPool = @(Get-SCMACAddressPool -VMHostGroup $HostGroup)
```

```
PS C:\> Get-SCMACAddress -MACAddressPool $MACAddressPool[0]
```

## Related topics

[Grant-SCMACAddress](#)

[Revoke-SCMACAddress](#)

---

# Get-SCMACAddressPool

---

## Get-SCMACAddressPool

Gets a MAC address pool.

### Syntax

Parameter Set: Default

```
Get-SCMACAddressPool [[-Name] <String> ] [-MACAddress <String> ] [-VirtualizationPlatform {Unknown | VirtualServer | HyperV | VMWareVC | VMWareESX | XENServer} ] [-VMHostGroup <HostGroup> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCMACAddressPool cmdlet gets one or more MAC address pools. A MAC address pool can be associated with one or more Virtual Machine Manager (VMM) host groups.

### Parameters

#### **-MACAddress<String>**

Specifies the MAC address or a set of MAC addresses for a physical or virtual network adapter on a computer.

Example format for a single MAC address:

```
-MACAddress "00-15-5D-B4-DC-00"
```

Example formats for a set of MAC addresses:

```
-MACAddress "00-15-5D-B4-DC-00", "00-1A-A0-E3-75-29"
```

```
$Macs = "00-15-5D-B4-DC-00", "00-1A-A0-E3-75-29"
```

```
Set-SCPXEserver -MACAddress $Macs
```

NOTE: When used with New-SCPXEserver or Set-SCPXEserver, the MACAddress parameter updates the PXE interfaces from which the SCDM PXE Server listens for and responds to PXE requests.

Aliases	none
Required?	false
Position?	named
Default Value	none



Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	false
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VirtualizationPlatform<VirtualizationPlatform>**

Specifies the virtualization platform of a virtual machine host managed by VMM. Valid values are: HyperV, VMWareESX, XENServer.

The acceptable values for this parameter are:

Unknown	
VirtualServer	
HyperV	
VMWareVC	
VMWareESX	
XENServer	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false

---

Accept Wildcard Characters?	false
-----------------------------	-------

### **-VMHostGroup<HostGroup>**

Specifies a virtual machine host group object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

### **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **MACAddressPool**

---

## Examples

### 1: Get all MAC address pools for a host group.

The first command gets the host group named "Production" and stores it in the \$HostGroup variable.

The second command gets all MAC address pools for the host group stored in \$HostGroup (including its parent host group if inheritance is enabled).

```
PS C:\> $HostGroup = Get-SCVMHostGroup | where { $_.Path -eq "All  
Hosts\HostGroup02\Production" }
```

```
PS C:\> Get-SCMACAddressPool -VMHostGroup $HostGroup
```

## Related topics

[New-SCMACAddressPool](#)

[Remove-SCMACAddressPool](#)

[Set-SCMACAddressPool](#)

---

# Get-SCNativeUplinkPortProfile

---

## Get-SCNativeUplinkPortProfile

Gets a native uplink port profile

### Syntax

Parameter Set: All

```
Get-SCNativeUplinkPortProfile [[-Name] <String> ] [-VMMServer <ServerConnection> ] [ <CommonParameters> ]
```

Parameter Set: ById

```
Get-SCNativeUplinkPortProfile [[-Name] <String> ] [-ID <Guid> ] [-VMMServer <ServerConnection> ] [ <CommonParameters> ]
```

### Detailed Description

The Get-SCNativeUplinkPortProfile cmdlet retrieves a native uplink port profile object.

### Parameters

#### **-ID<Guid>**

Specifies the numerical identifier (as a globally unique identifier, or GUID) for a specific object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

#### **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	false
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## **Examples**

### **1: Get a native uplink port profile by its name.**

This command gets the native uplink port profile object named NativeUplinkPortProfile01 and stores the object in the \$NativeUplinkPortProf variable.

```
PS C:\> $NativeUplinkPortProf = Get-SCNativeUplinkPortProfile -Name
"NativeUplinkPortProfile01"
```

## **Related topics**

[New-SCNativeUplinkPortProfile](#)

---

[Remove-SCNativeUplinkPortProfile](#)

[Set-SCNativeUplinkPortProfile](#)

---

# Get-SCNetworkGateway

---

## Get-SCNetworkGateway

Gets a network gateway object.

### Syntax

Parameter Set: Global

```
Get-SCNetworkGateway [-All] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ByID

```
Get-SCNetworkGateway -ID <Guid> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ByName

```
Get-SCNetworkGateway -Name <String> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCNetworkGateway cmdlet gets one or more network gateway objects. You can get an individual network gateway by its name or ID, or all network gateway objects.

### Parameters

#### -All

Retrieves a full list of all subordinate objects independent of the parent object. For example, the command Get-SCVirtualDiskDrive -All retrieves all virtual disk drive objects regardless of the virtual machine object or template object that each virtual disk drive object is associated with.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-ID<Guid>**

Specifies the numerical identifier (as a globally unique identifier, or GUID) for a specific object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false



---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

### Related topics

[Add-SCNetworkGateway](#)

[Read-SCNetworkGateway](#)

[Remove-SCNetworkGateway](#)

[Set-SCNetworkGateway](#)

[Test-SCNetworkGateway](#)

---

# Get-SCNotification

---

## Get-SCNotification

Gets update notifications for a service template or service instance.

### Syntax

Parameter Set: Default

```
Get-SCNotification [-NotifiedObject] <ClientObject> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCNotification cmdlet gets the update notifications for a service template or service instance. Update notifications alert you to updated resources that are available for a service template or service instance. Use the Set-SCNotifications cmdlet to dismiss notifications.

### Parameters

#### **-NotifiedObject<ClientObject>**

Specifies a service template object or service instance object for which you want to retrieve update notifications.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

#### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

---

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Examples

### 1: Get update notifications for a service.

The first command gets the service object named Service01 and stores the object in the \$Service variable.

The second command gets all update notifications for Service01.

```
PS C:\> $Service = Get-SCService -Name "Service01"
```

```
PS C:\> Get-SCNotification -NotifiedObject $Service
```

## Related topics

[Set-SCNotification](#)

---

# Get-SCOperatingSystem

---

## Get-SCOperatingSystem

Gets valid operating system objects from the VMM database.

### Syntax

Parameter Set: All

```
Get-SCOperatingSystem [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ID

```
Get-SCOperatingSystem -ID <Guid> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: Profile

```
Get-SCOperatingSystem -ApplicationProfile <ApplicationProfile> [-VMMServer  
<ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCOperatingSystem cmdlet gets one or more operating system objects from the Virtual Machine Manager (VMM) database. An operating system object is used to identify the operating system that is installed on a particular virtual hard disk.

### Parameters

#### **-ApplicationProfile<ApplicationProfile>**

Specifies an application profile object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

---

## **-ID<Guid>**

Specifies the numerical identifier (as a globally unique identifier, or GUID) for a specific object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **OperatingSystem**

---

## Examples

### 1: Get all operating system objects in your VMM environment.

This command gets all operating system objects from the VMM database on VMMServer01 and displays information about these operating system objects to the user.

```
PS C:\> Get-SCOperatingSystem -VMMServer "VMMServer01.Contoso.com"
```

### 2: Get all operating system objects in your VMM environment with the specified processor architecture.

This command gets all operating system objects from VMMServer01 and then selects only those operating systems that have an amd64 processor architecture. The command uses the Format-Table cmdlet to display only the Name and Architecture properties for each selected operating system.

```
PS C:\> Get-OperatingSystem -VMMServer "VMMServer01.Contoso.com" | where {$_.Architecture -eq "amd64"} | Format-Table -property Name,Architecture
```

## Related topics

[Set-SCVirtualHardDisk](#)

---

# Get-SCOpsMgrConnection

---

## Get-SCOpsMgrConnection

Gets the Operations Manager connection object.

### Syntax

Parameter Set: Default

```
Get-SCOpsMgrConnection [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCOpsMgrConnection cmdlet gets the Operations Manager connection object if a connection has been created.

For information about how to create an Operations Manager connection, type: "Get-Help New-SCOpsMgrConnection -detailed".

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

#### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).



---

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **OpsMgrConnection**

## Examples

### 1: Get the Operations Manager connection object.

The first command gets the Operations Manager connection object stores it in the \$OMConn variable.

The second command displays the Operations Manager connection object stored in \$OMConn to the user.

```
PS C:\> $OMConn = Get-SCOpsMgrConnection
```

```
PS C:\> $OMConn
```

## Related topics

[New-SCOpsMgrConnection](#)

[Remove-SCOpsMgrConnection](#)

[Set-SCOpsMgrConnection](#)

[Write-SCOpsMgrConnection](#)

---

# Get-SCPendingServiceSetting

---

## Get-SCPendingServiceSetting

Gets the pending service settings for a service that is in a pending service state.

### Syntax

Parameter Set: Default

```
Get-SCPendingServiceSetting [-Service] <Service> [-Name <String> ] [-VMMServer  
<ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCPendingServiceSetting cmdlet gets the pending service settings for a service that is in a pending service state.

### Parameters

#### **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

#### **-Service<Service>**

Specifies a VMM service object.

Aliases	none
---------	------

Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## **Examples**

### **1. Retrieve all pending service settings from a service in a pending service state**

The first command gets the service object named Service01 and stores the object in the \$Service variable.

The second command gets the pending service settings for Service01 and displays them for the user.

```
PS C:\> $Service = Get-SCService -Name "Service01"
```

```
PS C:\> Get-SCPendingServiceSetting -Service $Service
```

## **Related topics**

[Get-SCService](#)

---

# Get-SCPerformanceData

---

## Get-SCPerformanceData

Gets performance data for host groups, clusters, hosts, and virtual machines.

### Syntax

Parameter Set: Cluster

```
Get-SCPerformanceData [-VMHostCluster] <HostCluster> -PerformanceCounter <String> -TimeFrame <String> [ <CommonParameters>]
```

Parameter Set: Host

```
Get-SCPerformanceData [-VMHost] <Host> -PerformanceCounter <String> -TimeFrame <String> [ <CommonParameters>]
```

Parameter Set: HostGroup

```
Get-SCPerformanceData [-VMHostGroup] <HostGroup> -PerformanceCounter <String> -TimeFrame <String> [ <CommonParameters>]
```

Parameter Set: VM

```
Get-SCPerformanceData [-VM] <VM> -PerformanceCounter <String> -TimeFrame <String> [ <CommonParameters>]
```

## Detailed Description

The Get-SCPerformanceData cmdlet gets performance data for host groups, clusters, hosts, and virtual machines. You can request data for the following performance counters:

- CPU usage
- Memory usage
- Storage IOPS usage
- Network IO usage
- Power Savings

## Parameters

### **-PerformanceCounter<String>**

Specifies the performance counter to use. Valid values are: CPUUsage, MemoryUsage, StorageIOPSUsage, NetworkIOUsage, and PowerSavings.

---

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-TimeFrame<String>**

Specifies the timeframe in which to gather performance data. Valid values are: Hour, Day, Month.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VM<VM>**

Specifies a virtual machine object.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

---

## **-VMHost<Host>**

Specifies a virtual machine host object. VMM supports Hyper-V hosts, VMware ESX hosts, and Citrix XenServer hosts.

For more information about each type of host, type: "Get-Help Add-SCVMHost -detailed". See the examples for a specific cmdlet to determine how that cmdlet uses this parameter.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **-VMHostCluster<HostCluster>**

Specifies a VMM host cluster object.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **-VMHostGroup<HostGroup>**

Specifies a virtual machine host group object.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	True (ByValue)

---

Accept Wildcard Characters?	false
-----------------------------	-------

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- `double[]`

## Examples

### 1. Get performance data for a host group.

The first command gets the host group object named HostGroup01 and stores the object in the \$HostGroup variable.

The second command gets the memory usage data over the last three hours for HostGroup01.

```
PS C:\> $HostGroup = Get-SCVMHostGroup -Name "HostGroup01"
```

```
PS C:\> Get-SCPerformanceData -VMHostGroup $HostGroup -PerformanceCounter MemoryUsage -  
Timeframe Hour
```

### 2. Get performance data for a cluster.

The first command gets the cluster object named Cluster01 and stores the object in the \$Cluster variable.

The second command gets the memory usage data for the last day (24 hours) for Cluster01.

```
PS C:\> $Cluster = Get-SCVMHostCluster -Name "Cluster01.Contoso.com"
```

```
PS C:\> Get-SCPerformanceData -VMHostCluster $Cluster -PerformanceCounter MemoryUsage -  
Timeframe Day
```

### 3. Get performance data for a specified host.

The first command gets the host object named VMHost01 and stores the object in the \$VMHost variable.

The second command gets the memory usage data for the last month (30 days) for VMHost01.

```
PS C:\> $VMHost = Get-SCVMHost -ComputerName "VMHost01.Contoso.com"
```

```
PS C:\> Get-SCPerformanceData -VMHost $VMHost -PerformanceCounter MemoryUsage -Timeframe  
Month
```

---

## 4. Get performance data for a specified virtual machine.

The first command gets the virtual machine object named VM01 and stores the object in the \$VM variable.

The second command gets the memory usage data for the last day (24 hours) for VM01.

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM01"
```

```
PS C:\> Get-SCPerformanceData -VM $VM -PerformanceCounter MemoryUsage -Timeframe Day
```

### Related topics

[Get-SCVirtualMachine](#)

[Get-SCVMHost](#)

[Get-SCVMHostCluster](#)

[Get-SCVMHostGroup](#)



---

# Get-SCPlacementConfiguration

---

## Get-SCPlacementConfiguration

Gets the placement configuration settings for a host group.

### Syntax

Parameter Set: FromHostGroup

```
Get-SCPlacementConfiguration -VMHostGroup <HostGroup> [-VMMServer <ServerConnection> ] [ <CommonParameters> ]
```

### Detailed Description

The Get-SCPlacementConfiguration cmdlet gets the placement configuration settings that have been configured for a host group.

### Parameters

#### **-VMHostGroup<HostGroup>**

Specifies a virtual machine host group object or an array of host group objects.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

#### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
---------	------

---

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **PlacementConfiguration**

## Examples

### 1: Get the placement configuration for a host group.

The first command gets the host group object named HostGroup01 and stores the object in the \$HostGroup variable.

The second command gets the placement configuration object for the host group stored in \$HostGroup and places the object in the \$PlacementConfig variable.

The last command displays the placement configuration settings for the placement configuration stored in \$PlacementConfig for the user.

```
PS C:\> $HostGroup = Get-SCVMHostGroup "HostGroup01"
```

```
PS C:\> $PlacementConfig = Get-SCPlacementConfiguration -VMHostGroup $HostGroup
```

```
PS C:\> $PlacementConfig
```

## Related topics

[Get-SCVMHostGroup](#)

[Set-SCPlacementConfiguration](#)

---

# Get-SCPortClassification

---

## Get-SCPortClassification

Gets a port classification.

### Syntax

Parameter Set: All

```
Get-SCPortClassification [[-Name] <String> ] [-VMMServer <ServerConnection> ] [ <CommonParameters> ]
```

Parameter Set: ById

```
Get-SCPortClassification [[-Name] <String> ] [-ID <Guid> ] [-VMMServer <ServerConnection> ] [ <CommonParameters> ]
```

### Detailed Description

The Get-SCPortClassification cmdlet gets one or more port classification objects.

### Parameters

#### **-ID<Guid>**

Specifies the numerical identifier (as a globally unique identifier, or GUID) for a specific object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

#### **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	false
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## **Examples**

### **1. Get a port classification by its name.**

This command gets the port classification object named PortClass01.

```
PS C:\> Get-SCPortClassification -Name "PortClass01"
```

## **Related topics**

[New-SCPortClassification](#)

[Remove-SCPortClassification](#)

[Set-SCPortClassification](#)

---

# Get-SCPowerOptimizationRange

---

## Get-SCPowerOptimizationRange

Gets the set of time ranges when power optimization will be used.

### Syntax

Parameter Set: FromDOSettings

```
Get-SCPowerOptimizationRange -DynamicOptimizationConfiguration <HostGroupDOSettings> [-VMMServer <ServerConnection> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The Get-SCPowerOptimizationRange cmdlet gets the set of power optimization time ranges that have been added to a dynamic optimization configuration. During these time ranges, the hosts associated with the dynamic optimization configuration are turned on and off as needed.

### Parameters

#### **-DynamicOptimizationConfiguration<HostGroupDOSettings>**

Specifies a dynamic optimization configuration object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

#### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **-Confirm**

Prompts you for confirmation before executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- 
- **PowerOptimizationRange**

## Examples

### 1: Get the power optimization time ranges associated with a dynamic optimization configuration.

The first command gets the host group object named HostGroup01 and stores the object in the \$HostGroup variable.

The second command gets the dynamic optimization configuration object for the host group stored in \$HostGroup and stores the object in the \$DOConfig variable.

The third command gets the power optimization ranges that have been added to the dynamic optimization configuration stored in \$DOConfig.

The last command displays information about the power optimization ranges stored in \$PORange.

```
PS C:\> $HostGroup = Get-SCVMHostGroup "HostGroup01"
```

```
PS C:\> $DOConfig = Get-SCDynamicOptimizationConfiguration -VMHostGroup $HostGroup
```

```
PS C:\> $PORange = Get-SCPowerOptimizationRange -DynamicOptimizationConfiguration $DOConfig
```

```
PS C:\> $PORange
```

## Related topics

[Add-SCPowerOptimizationRange](#)

[Get-SCDynamicOptimizationConfiguration](#)

[Get-SCVMHostGroup](#)

---

# Get-SCPROMonitor

---

## Get-SCPROMonitor

Gets a PRO monitor object from Operations Manager.

### Syntax

Parameter Set: PROTargetType

```
Get-SCPROMonitor [-PROTargetType {VM | VMHost | VMMServer | ServiceInstance | Cloud | HostCluster | ComputerTier | Unspecified} ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: Name

```
Get-SCPROMonitor -ManagementPackName <String> -Name <String> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCPROMonitor cmdlet gets one or more Performance and Resource Optimization (PRO) monitors from Operations Manager.

### Parameters

#### **-ManagementPackName<String>**

Specifies the name of a management pack. To get a specific PRO monitor, use this parameter with the monitor's name property.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false



---

## **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-PROTargetType<PROTargetType>**

Specifies the PRO target type. Valid values are: Cloud, ComputerTier, HostCluster, ServiceInstance, Unspecified, VM, VMHost, VMMServer.

The acceptable values for this parameter are:

VM	
VMHost	
VMMServer	
ServiceInstance	
Cloud	
HostCluster	
ComputerTier	
Unspecified	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

### **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **PROMonitor**

### **Examples**

#### **1: Get all of the PRO monitors from Operations Manager.**

This command gets all PRO monitors from Operations Manager and displays details about each monitor for the user.

```
PS C:\> Get-SCPROMonitor
```

#### **2: Get a specific PRO monitor by its name.**

The first command gets the PRO monitor object with the specified name and management pack name, and stores the object in the \$PROMonitor variable.

The second command displays information about the monitor object stored in \$PROMonitor to the user.

```
PS C:\> $PROMonitor = Get-SCPROMonitor -Name "System Center Virtual Machine Manager Maximum  
Dynamic Memory Monitor" -ManagementPackName "System Center Virtual Machine Manager PRO V2  
HyperV Host Performance"
```

```
PS C:\> $PROMonitor
```

---

### 3: Get all PRO monitors for a specific target type.

This command gets all PRO monitors that have the target type of VMHost, and displays information about each monitor to the user.

```
PS C:\> Get-SCPROMonitor -PROTargetType VMHost
```

#### Related topics

[Get-SCOpsMgrConnection](#)

[Get-SCPROMonitorConfiguration](#)

[Get-SCPROMonitorState](#)

---

# Get-SCPRMonitorConfiguration

---

## Get-SCPRMonitorConfiguration

Gets a PRO monitor configuration object for a specified scope.

### Syntax

Parameter Set: HostGroup

```
Get-SCPRMonitorConfiguration -VMHostGroup <HostGroup> [-PROMonitor <PROMonitor> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: Cloud

```
Get-SCPRMonitorConfiguration -Cloud <Cloud> [-PROMonitor <PROMonitor> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ComputerTier

```
Get-SCPRMonitorConfiguration -ComputerTier <ComputerTier> [-PROMonitor <PROMonitor> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: HostCluster

```
Get-SCPRMonitorConfiguration -VMHostCluster <HostCluster> [-PROMonitor <PROMonitor> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: Service

```
Get-SCPRMonitorConfiguration -Service <Service> [-PROMonitor <PROMonitor> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: VM

```
Get-SCPRMonitorConfiguration -VM <VM> [-PROMonitor <PROMonitor> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: VMHost

```
Get-SCPRMonitorConfiguration -VMHost <Host> [-PROMonitor <PROMonitor> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: VMMServer

```
Get-SCPRMonitorConfiguration -VMMServerScope [-PROMonitor <PROMonitor> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCPRMonitorConfiguration cmdlet gets one or more Performance and Resource Optimization (PRO) monitor configuration objects for a specified scope.

---

## Parameters

### **-Cloud<Cloud>**

Specifies a private cloud object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-ComputerTier<ComputerTier>**

Specifies a computer tier object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-PROMonitor<PROMonitor>**

Specifies a PRO monitor object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-Service<Service>**

Specifies a VMM service object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VM<VM>**

Specifies a virtual machine object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMHost<Host>**

Specifies a virtual machine host object. VMM supports Hyper-V hosts, VMware ESX hosts, and Citrix XenServer hosts.

For more information about each type of host, type: "Get-Help Add-SCVMHost -detailed". See the examples for a specific cmdlet to determine how that cmdlet uses this parameter.

Aliases	none
Required?	true
Position?	named
Default Value	none

---

Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMHostCluster<HostCluster>**

Specifies a VMM host cluster object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMHostGroup<HostGroup>**

Specifies a virtual machine host group object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **-VMMServerScope**

Indicates that the PRO information returned is scoped to the entire VMM server.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **PROMonitorConfiguration**

## **Examples**

### **1: Get all PRO monitor configuration objects on a specified host group.**

The first command gets the host group object named Production and stores the object in the \$HostGroup variable.

The second command gets the PRO monitor configuration objects on the host group stored in \$HostGroup (Production) and stores the objects in the \$PROMonitorConfig variable.

The last command displays information about the PRO monitor configuration objects stored in \$PROMonitorConfig to the user.

```
PS C:\> $HostGroup = Get-SCVMHostGroup -Name "Production"
```

```
PS C:\> $PROMonitorConfig = Get-SCPROMonitorConfiguration -VMHostGroup $HostGroup
```

```
PS C:\> $PROMonitorConfig
```



---

## Related topics

[Set-SCPROMonitorConfiguration](#)

---

# Get-SCPROMonitorState

---

## Get-SCPROMonitorState

Gets the state of a specified PRO monitor on a specific VMM object.

### Syntax

Parameter Set: VMHost

```
Get-SCPROMonitorState -VMHost <Host> [-PROMonitor <PROMonitor> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: Cloud

```
Get-SCPROMonitorState -Cloud <Cloud> [-PROMonitor <PROMonitor> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ComputerTier

```
Get-SCPROMonitorState -ComputerTier <ComputerTier> [-PROMonitor <PROMonitor> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: HostCluster

```
Get-SCPROMonitorState -VMHostCluster <HostCluster> [-PROMonitor <PROMonitor> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: Service

```
Get-SCPROMonitorState -Service <Service> [-PROMonitor <PROMonitor> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: VM

```
Get-SCPROMonitorState -VM <VM> [-PROMonitor <PROMonitor> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: VMMServer

```
Get-SCPROMonitorState -VMMServerScope [-PROMonitor <PROMonitor> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCPROMonitorState cmdlet gets the state of a specified Performance and Resource Optimization (PRO) monitor on a specific Virtual Machine Manager (VMM) object.

---

## Parameters

### **-Cloud<Cloud>**

Specifies a private cloud object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-ComputerTier<ComputerTier>**

Specifies a computer tier object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-PROMonitor<PROMonitor>**

Specifies a PRO monitor object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

---

## **-Service<Service>**

Specifies a VMM service object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VM<VM>**

Specifies a virtual machine object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMHost<Host>**

Specifies a virtual machine host object. VMM supports Hyper-V hosts, VMware ESX hosts, and Citrix XenServer hosts.

For more information about each type of host, type: "Get-Help Add-SCVMHost -detailed". See the examples for a specific cmdlet to determine how that cmdlet uses this parameter.

Aliases	none
Required?	true
Position?	named
Default Value	none

---

Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMHostCluster<HostCluster>**

Specifies a VMM host cluster object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-VMMServerScope**

Indicates that the PRO information returned is scoped to the entire VMM server.

Aliases	none
Required?	true
Position?	named
Default Value	none

---

Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **PROMonitorState**

## Examples

### 1: Get the state of a PRO monitor for a specific host.

The first command gets the virtual machine host object named VMHost01 and stores the object in the \$VMHost variable.

The second command gets the PRO monitor object with the specified name and management pack name and stores the object in the \$PROMonitor variable.

The third command gets the PRO monitor state for the PRO monitor object stored in \$PROMonitor for VMHost01 and stores the state in the \$PROMonitorState variable.

The last command displays the state information stored in \$PROMonitorState for the user.

```
PS C:\> $VMHost = Get-VMHost "VMHost01.Contoso.com"
```

```
PS C:\> $PROMonitor = Get-SCPROMonitor -Name "System Center Virtual Machine Manager Maximum  
Dynamic Memory Monitor" -ManagementPackName "System Center Virtual Machine Manager PRO V2  
HyperV Host Performance"
```

```
PS C:\> $PROMonitorState = Get-SCPROMonitorState -PROMonitor $PROMonitor -VMHost $VMHost
```

```
PS C:\> $PROMonitorState
```

## Related topics

[Get-SCPROMonitor](#)

[Reset-SCPROMonitorState](#)

---

# Get-SCPROTip

---

## Get-SCPROTip

Gets PRO tip objects from the VMM database.

### Syntax

Parameter Set: AllTips

```
Get-SCPROTip [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: CloudTips

```
Get-SCPROTip -Cloud <Cloud> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ClusterTips

```
Get-SCPROTip -VMHostCluster <HostCluster> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: GetById

```
Get-SCPROTip -PROTipID <Guid> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: HostTips

```
Get-SCPROTip -VMHost <Host> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ServiceTips

```
Get-SCPROTip -Service <Service> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: VMMServerTips

```
Get-SCPROTip -VMMServerScope [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: VMTips

```
Get-SCPROTip -VM <VM> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

## Detailed Description

The Get-SCPROTip cmdlet gets one or more Performance and Resource Optimization (PRO) tip objects from the Virtual Machine Manager (VMM) database.

If PRO is enabled, a PRO tip recommends an action in response to an alert generated by Operations Manager for hosts that are members of a host group or for hosts configured in a host cluster, as well as for the virtual machines deployed on those hosts. A recommended action might be to move a virtual machine to a new host or to add a CPU to a virtual machine.

PRO provides workload and application-aware resource optimization within host groups or host clusters that are managed by both VMM and Operations Manager. To receive PRO tips for these hosts, you

---

must first configure PRO for VMM. This includes deploying Operations Manager, which generates the PRO tips based on monitors provided by PRO-enabled management packs. PRO tip recommendations are based on policies implemented through Operations Manager.

You can use Get-SCPROTip and the other PROTip cmdlets only on Hyper-V, VMware, or Citrix XenServer hosts that belong to a host group, that are configured in a host cluster, or that belong to a private cloud.

## Parameters

### **-Cloud<Cloud>**

Specifies a private cloud object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-Service<Service>**

Specifies a VMM service object.

Aliases	none
---------	------



Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-VM<VM>**

Specifies a virtual machine object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-VMHost<Host>**

Specifies a virtual machine host object. VMM supports Hyper-V hosts, VMware ESX hosts, and Citrix XenServer hosts.

For more information about each type of host, type: "Get-Help Add-SCVMHost -detailed". See the examples for a specific cmdlet to determine how that cmdlet uses this parameter.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-VMHostCluster<HostCluster>**

Specifies a VMM host cluster object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-VMMServerScope**

Indicates that the PRO information returned is scoped to the entire VMM server.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- PROTip

## Examples

### 1: Get all PRO tips for all hosts and all virtual machines managed by VMM.

This command gets all PRO tips from the VMM database and displays information about each PRO tip to the user.

```
PS C:\> Get-SCPROTip
```

### 2: Get all active PRO tips for a specific host.

The first command gets the host object named VMHost01 and stores the object in the \$VMHost variable.

The second command gets all active PRO tips for VMHost01 and displays information about each tip to the user.

```
PS C:\> $VMHost = Get-SCVMHost -ComputerName "VMHost01.Contoso.com"
```

```
PS C:\> Get-SCPROTip -VMHost $VMHost
```

## Related topics

[Clear-SCPROTip](#)

[Invoke-SCPROTip](#)

[Set-SCPROTip](#)

[Test-SCPROTip](#)

---

# Get-SCPXESever

---

## Get-SCPXESever

Gets a PXESever object from the VMM database.

### Syntax

Parameter Set: ID

```
Get-SCPXESever [-ID <Guid> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: Name

```
Get-SCPXESever [-ComputerName <String> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCPXESever cmdlet gets one or more PXESever objects from the Virtual Machine Manager (VMM) database.

For information about adding a PXESever object to VMM, type: "Get-Help Add-SCPXESever -detailed".

### Parameters

#### **-ComputerName<String>**

Specifies the name of a computer that VMM can uniquely identify on your network. Valid formats are: FQDN, IPv4 or IPv6 address, or NetBIOS name.

NOTE: See the examples for a specific cmdlet to determine how that cmdlet specifies the computer name.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-ID<Guid>**

Specifies the numerical identifier (as a globally unique identifier, or GUID) for a specific object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **PXE Server**

---

## Examples

### 1: Retrieve a PXE server by its FQDN.

This command gets the PXE server named WDSserver01.

```
PS C:\> Get-SCPXEserver -ComputerName "WDSserver01.Contoso.com"
```

## Related topics

[Add-SCPXEserver](#)

[Remove-SCPXEserver](#)

---

# Get-SCRunAsAccount

---

## Get-SCRunAsAccount

Gets VMM Run As accounts.

### Syntax

Parameter Set: Default

```
Get-SCRunAsAccount [[-Name] <String> ] [-ID <Guid> ] [-IsEnabled <Boolean> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCRunAsAccount cmdlet gets Virtual Machine Manager (VMM) Run As accounts.

### Parameters

#### -ID<Guid>

Specifies the numerical identifier (as a globally unique identifier, or GUID) for a specific object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

#### -IsEnabled<Boolean>

Retrieves, when set to \$True, Run As accounts that are enabled. When set to \$False, Run Accounts that are disabled are returned.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	false
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).



---

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **RunAsAccount**
- **RunAsAccount[]**

## Examples

### 1: Get a Run As account by its name.

The first command gets the Run As account object named RunAsAccount01 and stores the object in the \$RunAsAccount variable.

The second command displays information about the Run As account stored in \$RunAsAccount to the user.

```
PS C:\> $RunAsAccount = Get-SCRunAsAccount -Name "RunAsAccount01"
PS C:\> $RunAsAccount
```

### 2: Get an enabled Run As account by its name.

The first command gets the enabled Run As account object named RunAsAccount01 and stores the object in the \$RunAsAccount variable.

The second command displays information about the Run As account stored in \$RunAsAccount to the user.

```
PS C:\> $RunAsAccount = Get-SCRunAsAccount -Name "RunAsAccount01" -IsEnabled $True
PS C:\> $RunAsAccount
```

### 3: Get enabled Run As accounts that contain a specified string in their name.

The first command gets all enabled Run As account objects that contain "Account" in their names and stores the objects in the \$RunAsAccount array.

The second command displays information about the first Run As account in the \$RunAsAccount array to the user. The third command displays information about the second Run As account in the \$RunAsAccount array to the user.

```
PS C:\> $RunAsAccount = Get-SCRunAsAccount -Name *Account* -IsEnabled $True
PS C:\> $RunAsAccount[0]
PS C:\> $RunAsAccount[1]
```

## Related topics

[New-SCRunAsAccount](#)

---

[Remove-SCRunAsAccount](#)

[Disable-SCRunAsAccount](#)

[Enable-SCRunAsAccount](#)

[Set-SCRunAsAccount](#)

---

# Get-SCRunAsAccountConsumer

---

## Get-SCRunAsAccountConsumer

Gets the Run As account consumer objects for a specified Run As account.

### Syntax

Parameter Set: Default

```
Get-SCRunAsAccountConsumer [[-RunAsAccount] <RunAsAccount> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCRunAsAccountConsumer cmdlet gets the Run As account consumer objects for a specified Run As account. Get-SCRunAsAccountConsumer returns any Virtual Machine Manager (VMM) object that refers to the given Run As account.

### Parameters

#### **-RunAsAccount<RunAsAccount>**

Specifies a Run As account that contains credentials with permission to perform this action.

Aliases	none
Required?	false
Position?	1
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

#### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

---

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Examples

### 1. Get the load balancers that use a specified Run As account.

The first command gets the Run As account object named LBRunAsAcct01 and stores the object in the \$RunAsAcct variable.

The second command gets the Run As account consumer objects for the load balancers that use the Run As account stored in \$RunAsAcct and stores the consumer objects in the \$RAAConsumers variable.

The last command displays the Run As account consumer objects stored in \$RAAConsumers to the user.

```
PS C:\> $RunAsAcct = Get-SCRunAsAccount -Name "LBRunAsAcct01"
PS C:\> $RAAConsumers = Get-SCRunAsAccountConsumer -RunAsAccount $RunAsAcct
PS C:\> $RAAConsumers
```

## Related topics

[Get-SCRunAsAccount](#)

---

# Get-SCScript

---

## Get-SCScript

Gets script objects from the VMM library, which allows you to view or edit any script, or to view, edit, or run a Windows PowerShell script (if you have appropriate permissions).

### Syntax

Parameter Set: All

```
Get-SCScript [-All] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: EquivalentResourceParamSet

```
Get-SCScript -FamilyName <String> [-Release <String> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ID

```
Get-SCScript [-ID <Guid> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: NameParamSet

```
Get-SCScript -Name <String> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCScript cmdlet gets script objects from the Virtual Machine Manager (VMM) library. The script file that a script object represents is stored in the file system on a library server. Typically, these scripts are either Windows PowerShell scripts or answer file scripts (including Sysprep.inf and Unattend.xml files, which contain the inputs required for the Windows Setup program).

As illustrated in the examples, you can use Get-SCScript not only to retrieve script objects but also (if you have appropriate permissions) to view the contents of a script or to edit a script. In addition, you can run the script if the following are true: 1) the script is a Windows PowerShell script, 2) scripting is enabled on your server, and 3) you have appropriate permissions (see example 5).

For information about enabling Windows PowerShell scripting on your server, type: "Get-Help about\_Signing", "Get-Help Get-ExecutionPolicy -detailed", and "Get-Help Set-ExecutionPolicy -detailed".

---

## Parameters

### **-All**

Retrieves a full list of all subordinate objects independent of the parent object. For example, the command `Get-SCVirtualDiskDrive -All` retrieves all virtual disk drive objects regardless of the virtual machine object or template object that each virtual disk drive object is associated with.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-FamilyName<String>**

Specifies a family name for a physical resource in the VMM library. This value is used in conjunction with `Release`, `Namespace`, and `Type` to establish equivalency among library resources.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-ID<Guid>**

Specifies the numerical identifier (as a globally unique identifier, or GUID) for a specific object.

Aliases	none
Required?	false
Position?	named
Default Value	none

---

Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Release<String>**

Specifies a string that describes the release of a library resource. VMM automatically creates a release value for every resource imported into the library. After the resource has been imported, the string can be customized.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false

---

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **Script**

## Examples

### 1: Get all scripts stored on all VMM library servers.

This command gets all script objects stored in library shares in the VMM library on VMMServer01, and then displays information about these scripts to the user.

```
PS C:\> Get-SCScript -VMMServer "VMMServer01.Contoso.com"
```

### 2: Display specified information about all scripts on a library server.

This command gets all script objects stored on LibraryServer01 and displays the name, library server, and share path for these scripts to the user.

```
PS C:\> Get-SCScript -VMMServer "VMMServer01.Contoso.com" | where { $_.LibraryServer.Name -eq "LibraryServer01.Contoso.com" } | Format-List -Property Name, LibraryServer, SharePath
```

### 3: Get all scripts with a specific name on any VMM library server.

This command gets the answer file script objects named Sysprep.inf that are stored on any library server on VMMServer01.

NOTE: By default, the name of a script object in the VMM library is the same name (including the file extension) as the name of the actual script file on the library server.

```
PS C:\> Get-SCScript -VMMServer "VMMServer1.Contoso.com" | where { $_.Name -eq "Sysprep.inf" }
```



---

## 4: View a script that is stored in the VMM library.

The first command gets the script object named "SummarizeVMMInfo.ps1" from the VMM library and stores the object in the \$Script variable.

The second command uses Notepad to open the script so that you can view its contents (if you have the appropriate permissions to read the script).

NOTE: If you have appropriate write permissions, you can also edit the script and save the new version.

```
PS C:\> $Script = Get-SCScript | where { $_.Name -eq "SummarizeVMMInfo.ps1" }
```

```
PS C:\> Notepad.exe $Script.SharePath
```

## 5: Run a Windows PowerShell script that is stored in the VMM library.

The first command gets the script object named "SummarizeVMMInfo.ps1" from the VMM library and stores the object in the \$Script variable.

The second command uses the "&" operator to run the script stored in \$Script.

To run a Windows PowerShell script stored in a VMM library share, you must ensure the following:

- You have read and execute permissions on the script file.
- You are member of the VMM Administrators user role.
- You have permissions to access the VMM library share.
- Windows PowerShell scripting is enabled. If it isn't:
  1. Run the VMM command shell as an Administrator.
  2. Use the Set-ExecutionPolicy cmdlet to set the execution policy to the appropriate level for your environment.

For more information, type:

```
Get-Help about_Signing
```

```
Get-Help Get-ExecutionPolicy -detailed
```

```
Get-Help Set-ExecutionPolicy -detailed
```

```
PS C:\> $Script = Get-SCScript | where { $_.Name -eq "SummarizeVMMInfo.ps1" }
```

```
PS C:\> &$Script.SharePath
```

## Related topics

[Remove-SCScript](#)

[Set-SCScript](#)

---

# Get-SCScriptCommand

---

## Get-SCScriptCommand

Gets all script commands for an application profile, application deployment, or host profile.

### Syntax

Parameter Set: ApplicationDeployment

```
Get-SCScriptCommand -ApplicationDeployment <ApplicationDeployment> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ApplicationProfile

```
Get-SCScriptCommand -ApplicationProfile <ApplicationProfile> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: VMHostProfile

```
Get-SCScriptCommand -VMHostProfile <VMHostProfile> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCScriptCommand cmdlet gets all script commands for an application profile, application deployment, or host profile.

### Parameters

#### **-ApplicationDeployment<ApplicationDeployment>**

Specifies an application deployment object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

---

## **-ApplicationProfile<ApplicationProfile>**

Specifies an application profile object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **-VMHostProfile<VMHostProfile>**

Specifies a virtual machine host profile object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **ScriptCommand**

## Examples

### 1: Get all script commands associated with an application profile.

The first command gets the application profile object named SvcWebAppProfile01 and stores the object in the \$AppProfile variable.

The second command gets all script commands for the application profile object stored in \$AppProfile and stores the objects in the \$ScriptCommand array.

The last command displays information about all of the script command objects stored in the \$ScriptCommand array to the user.

```
PS C:\> $AppProfile = Get-SCApplicationProfile -Name "SvcWebAppProfile01"
```

```
PS C:\> $ScriptCommand = Get-SCScriptCommand -ApplicationProfile $AppProfile
```

```
PS C:\> $ScriptCommand
```

### 2: Get all of the script commands associated with an application deployment

The first command gets the application profile object named SvcWebAppProfile01 and stores the object in the \$AppProfile variable.

The second command gets the application deployment object for the application profile stored in \$AppProfile and stores the object in the \$AppDeployment variable.

The last command gets all of the script commands associated with the application deployment object stored in \$AppDeployment and stores the script commands in \$ScriptCommand.

```
PS C:\> $AppProfile = Get-SCApplicationProfile -Name "SvcWebAppProfile01"
```

```
PS C:\> $AppDeployment = Get-SCApplicationDeployment -ApplicationProfile $AppProfile
```

```
PS C:\> $ScriptCommand = Get-SCScriptCommand -ApplicationDeployment $AppDeployment
```

## Related topics

[Add-SCScriptCommand](#)

[Remove-SCScriptCommand](#)

[Set-SCScriptCommand](#)

---

# Get-SCScriptCommandSetting

---

## Get-SCScriptCommandSetting

Gets the settings for a script command.

### Syntax

Parameter Set: Default

```
Get-SCScriptCommandSetting -ScriptCommand <SCScriptCommand> [ <CommonParameters>]
```

### Detailed Description

The Get-SCScriptCommandSetting cmdlet gets the settings that have been configured on a script command.

### Parameters

#### **-ScriptCommand<SCScriptCommand>**

Specifies a script command object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

#### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

---

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **ScriptCommandSetting**

## Examples

### 1: Get the script command settings for a specific script command.

The first command gets the application profile object named SvcWebAppProfile01 and stores the object in the \$AppProfile variable.

The second command gets the script command object named PostInstall and stores the object in the \$ScriptCommand variable.

The last command gets the script command settings for the script command stored in \$ScriptCommand and displays the settings for the user.

```
PS C:\> $AppProfile = Get-SCApplicationProfile -Name "SvcWebAppProfile01"
```

```
PS C:\> $ScriptCommand = Get-SCScriptCommand -ApplicationProfile $AppProfile | Where  
{$_ .Name -eq "PostInstall"}
```

```
PS C:\> Get-SCScriptCommandSetting -ScriptCommand $ScriptCommand
```

## Related topics

[New-SCScriptCommandSetting](#)

[Set-SCScriptCommandSetting](#)

---

# Get-SCServerFeature

---

## Get-SCServerFeature

Gets the operating system roles and features that have been added to a guest OS profile.

### Syntax

Parameter Set: AllParamSet

```
Get-SCServerFeature [-Name <String> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: IDParamSet

```
Get-SCServerFeature -ID <Guid> [-Name <String> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: OSParamSet

```
Get-SCServerFeature -OperatingSystem <OperatingSystem> [-Name <String> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: OSProfileParamSet

```
Get-SCServerFeature -GuestOSProfile <GuestOSProfile> [-Name <String> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: TemplateParamSet

```
Get-SCServerFeature -VMTemplate <Template> [-Name <String> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

## Detailed Description

The Get-SCServerFeature cmdlet gets the operating system roles and features that have been added to a guest OS profile.

## Parameters

### **-GuestOSProfile<GuestOSProfile>**

Specifies a guest operating system profile object.

Aliases	none
Required?	true

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-ID<Guid>**

Specifies the numerical identifier (as a globally unique identifier, or GUID) for a specific object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-OperatingSystem<OperatingSystem>**

Specifies the type of operating system for a virtual machine. To list the names of all available operating systems in VMM, type: "Get-SCOperatingSystem".

Aliases	none
---------	------



Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-VMTemplate<Template>**

Specifies a VMM template object used to create virtual machines.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

---

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **ServerFeature**

## Examples

### 1: Get all operating system features than have been added to a guest OS profile.

The first command gets the guest OS profile object named NewOSProfile01 and stores the object in the \$OSProfile variable.

The second command gets all of the server feature objects that have been added to the guest OS profile stored in \$OSProfile and stores the objects in the \$ServerFeature array.

```
PS C:\> $OSProfile = Get-SCGuestOSProfile -Name "NewOSProfile01"
```

```
PS C:\> $ServerFeature = Get-SCServerFeature -GuestOSProfile $OSProfile
```

## Related topics

[Add-SCServerFeature](#)

[Get-SCGuestOSProfile](#)

[Remove-SCServerFeature](#)

---

# Get-SCService

---

## Get-SCService

Gets a VMM service.

### Syntax

Parameter Set: All

```
Get-SCService [-All] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: Cloud

```
Get-SCService -Cloud <Cloud> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ID

```
Get-SCService [-ID <Guid> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: Name

```
Get-SCService [[-Name] <String> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCService cmdlet gets one or more services in Virtual Machine Manager (VMM).

### Parameters

#### -All

Retrieves a full list of all subordinate objects independent of the parent object. For example, the command Get-SCVirtualDiskDrive -All retrieves all virtual disk drive objects regardless of the virtual machine object or template object that each virtual disk drive object is associated with.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-Cloud<Cloud>**

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-ID<Guid>**

Specifies the numerical identifier (as a globally unique identifier, or GUID) for a specific object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	false
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **Service**

## **Examples**

### **1: Get all services.**

This command gets all services and then displays the values for their name and ID properties in a table format.

```
PS C:\> Get-SCService | Format-Table -property Name,ID
```

## **Related topics**

[New-SCService](#)

[Read-SCService](#)

[Remove-SCService](#)

[Resume-SCService](#)

[Set-SCService](#)

[Start-SCService](#)

[Stop-SCService](#)

[Suspend-SCService](#)

---

[Update-SCService](#)

---

# Get-SCServiceConfiguration

---

## Get-SCServiceConfiguration

Gets a service configuration object stored in the VMM library.

### Syntax

Parameter Set: Default

```
Get-SCServiceConfiguration [-ID <Guid> ] [-Name <String> ] [-VMMServer <ServerConnection> ]  
[ <CommonParameters>]
```

### Detailed Description

The Get-SCServiceConfiguration cmdlet gets one or more service configuration objects stored in the Virtual Machine Manager (VMM) library.

### Parameters

#### **-ID<Guid>**

Specifies the numerical identifier (as a globally unique identifier, or GUID) for a specific object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

#### **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **ServiceConfiguration**

## **Examples**

### **1: Get all service configuration objects in the library.**

The first command gets all service configuration objects on VMMServer01, selects from the results the service configuration objects that contain "Service" in their name, and then stores the objects in the \$SvcConfigs variable.

The second command displays the properties of the service configuration objects to the user.

```
PS C:\> $SvcConfigs = Get-SCServiceConfiguration -VMMServer "VMMServer01.Contoso.com" |
where { $_.Name -match "Service" }
```



---

PS C:\> \$SvcConfigs

## Related topics

[New-SCServiceConfiguration](#)

[Remove-SCServiceConfiguration](#)

[Set-SCServiceConfiguration](#)

[Update-SCServiceConfiguration](#)

---

# Get-SCServiceSetting

---

## Get-SCServiceSetting

Gets a service setting for a service template or a service instance.

### Syntax

Parameter Set: Service

```
Get-SCServiceSetting -Service <Service> [-Name <String> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ServiceConfiguration

```
Get-SCServiceSetting -ServiceConfiguration <ServiceConfiguration> [-Name <String> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ServiceTemplate

```
Get-SCServiceSetting -ServiceTemplate <ServiceTemplate> [-Name <String> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

## Detailed Description

Get-SCServiceSetting cmdlet gets one or more service settings for a service template or a service instance.

## Parameters

### -Name<String>

Specifies the name of a VMM object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-Service<Service>**

Specifies a VMM service object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **-ServiceConfiguration<ServiceConfiguration>**

Specifies a service configuration object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **-ServiceTemplate<ServiceTemplate>**

Specifies a service template object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

---

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Examples**

### **1. Retrieve all service settings from a service template.**

The first command gets the service template object named ServiceTemplate01 with the release of Beta and stores the object in the \$Template variable.

The second command gets all service settings for ServiceTemplate01 and displays their properties to the user.

```
PS C:\> $Template = Get-SCServiceTemplate -Name "ServiceTemplate01" | where {$_.Release -eq "Beta"}
PS C:\> Get-SCServiceSetting -ServiceTemplate $Template
```

### **2. Retrieve a service setting from a service configuration.**

The first command gets the service configuration object named ServiceConfig01 and stores the object in the \$Config variable.

The second command gets the service setting object named Setting01 associated with ServiceConfig01 and stores the object in the \$ServiceSetting variable.

The last command displays the properties for the service setting stored in \$ServiceSetting.

```
PS C:\> $Config = Get-SCServiceConfiguration -Name "ServiceConfig01"
PS C:\> $Setting = Get-SCServiceSetting -ServiceConfiguration $Config -Name "Setting01"
PS C:\> $Setting
```

---

## Related topics

[Set-SCServiceSetting](#)

---

# Get-SCServiceTemplate

---

## Get-SCServiceTemplate

Gets a service template stored in the VMM library.

### Syntax

Parameter Set: All

```
Get-SCServiceTemplate [[-Name] <String> ] [-All] [-VMMServer <ServerConnection> ] [  
<CommonParameters>]
```

Parameter Set: ID

```
Get-SCServiceTemplate [[-Name] <String> ] [-ID <Guid> ] [-VMMServer <ServerConnection> ] [  
<CommonParameters>]
```

### Detailed Description

The Get-SCServiceTemplate cmdlet gets one or more service templates stored in the Virtual Machine Manager (VMM) library.

### Parameters

#### -All

Retrieves a full list of all subordinate objects independent of the parent object. For example, the command Get-SCVirtualDiskDrive -All retrieves all virtual disk drive objects regardless of the virtual machine object or template object that each virtual disk drive object is associated with.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-ID<Guid>**

Specifies the numerical identifier (as a globally unique identifier, or GUID) for a specific object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	false
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **ServiceTemplate**

## Examples

### 1: Get all service template objects in the library.

The first command gets all service template objects on VMMServer01, selects from the results the service template objects that contain "Service" in their name, and then stores the objects in the \$SvcTemplates variable.

The second command displays information about the service template objects to the user.

```
PS C:\> $SvcTemplates = Get-SCServiceTemplate -VMMServer "VMMServer01.Contoso.com" | where {
$_.Name -match "Service" }
```

```
PS C:\> $SvcTemplates
```

## Related topics

[New-SCServiceTemplate](#)

[Read-SCServiceTemplate](#)

[Remove-SCServiceTemplate](#)

[Resolve-SCServiceTemplate](#)

[Set-SCServiceTemplate](#)

[Test-SCServiceTemplate](#)



---

# Get-SCServicingWindow

---

## Get-SCServicingWindow

Gets a list of servicing windows that are assigned to a virtual machine, a host, or a service.

### Syntax

Parameter Set: Connection

```
Get-SCServicingWindow [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: FromID

```
Get-SCServicingWindow -ID <Guid> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: FromName

```
Get-SCServicingWindow [-Name] <String> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: FromService

```
Get-SCServicingWindow -Service <Service> [-ServicingWindowFilter {Now | Next | All} ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: FromVM

```
Get-SCServicingWindow -VM <VM> [-ServicingWindowFilter {Now | Next | All} ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: FromVMHost

```
Get-SCServicingWindow -VMHost <Host> [-ServicingWindowFilter {Now | Next | All} ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

## Detailed Description

The Get-SCServicingWindow cmdlet gets a list of servicing windows that are assigned to a virtual machine, a host, or a service.

## Parameters

### **-ID<Guid>**

Specifies the numerical identifier (as a globally unique identifier, or GUID) for a specific object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Service<Service>**

Specifies a VMM service object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-ServicingWindowFilter<ServicingWindowFilterType>**

Specifies a filter for retrieving servicing windows. Valid values are: Now, Next, All.

The acceptable values for this parameter are:

Now	
Next	
All	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VM<VM>**

Specifies a virtual machine object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMHost<Host>**

Specifies a virtual machine host object. VMM supports Hyper-V hosts, VMware ESX hosts, and Citrix XenServer hosts.

For more information about each type of host, type: "Get-Help Add-SCVMHost -detailed". See the examples for a specific cmdlet to determine how that cmdlet uses this parameter.

Aliases	none
Required?	true

Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **ServicingWindow**

## **Examples**

### **1: Get a specific servicing window by its name.**

The first command gets the servicing window object named Backup Staging A and stores the object in the \$SvcWindow variable.

The second command displays the information about the servicing window stored in \$SvcWindow (Backup Staging A) to the user.

```
PS C:\> $SvcWindow = Get-SCServicingWindow -Name "Backup Staging A"
```

```
PS C:\> $SvcWindow
```

---

## Related topics

[New-SCServicingWindow](#)

[Remove-SCServicingWindow](#)

[Set-SCServicingWindow](#)

---

# Get-SCServicingWindowSubscription

---

## Get-SCServicingWindowSubscription

Gets a list of servicing window subscriptions.

### Syntax

Parameter Set: Connection

```
Get-SCServicingWindowSubscription [[-ServicingWindow] <ServicingWindow> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: FromHost

```
Get-SCServicingWindowSubscription [[-ServicingWindow] <ServicingWindow> ] -VMHost <Host> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: FromService

```
Get-SCServicingWindowSubscription [[-ServicingWindow] <ServicingWindow> ] -Service <Service> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: FromVM

```
Get-SCServicingWindowSubscription [[-ServicingWindow] <ServicingWindow> ] -VM <VM> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCServicingWindowSubscription cmdlet gets a list of servicing window subscriptions.

### Parameters

#### **-Service<Service>**

Specifies a VMM service object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false

Accept Wildcard Characters?	false
-----------------------------	-------

## **-ServicingWindow<ServicingWindow>**

Specifies a servicing window object.

Aliases	none
Required?	false
Position?	1
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **-VM<VM>**

Specifies a virtual machine object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMHost<Host>**

Specifies a virtual machine host object. VMM supports Hyper-V hosts, VMware ESX hosts, and Citrix XenServer hosts.

For more information about each type of host, type: "Get-Help Add-SCVMHost -detailed". See the examples for a specific cmdlet to determine how that cmdlet uses this parameter.

Aliases	none
Required?	true
Position?	named

Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **ServiceWindowSubscription**

## **Examples**

### **1: Get a list of a servicing window subscriptions for a virtual machine.**

The first command gets the virtual machine object named VM01 and stores the object in the \$VM variable.

The second command gets the servicing window subscription objects for the virtual machine stored in \$VM (VM01) and stores the objects in the \$SvcWindowSub variable.

The last command displays the servicing windows stored in \$SvcWindowSub for the user.

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM01"
```

```
PS C:\> $SvcWindowSub = Get-SCServicingWindowSubscription -VM $VM
```



---

PS C:\> \$SvcWindowSub

## 2: Get a list of subscriptions for a specified servicing window.

The first command gets the servicing window object named Backup Staging A and stores the object in the \$SvcWindow variable.

The second command gets the servicing window subscription objects for the servicing window stored in \$SvcWindow (Backup Staging A) and stores the objects in the \$SvcWindowSub variable.

The last command displays the list of subscriptions stored in \$SvcWindowSub to the user.

```
PS C:\> $SvcWindow = Get-SCServicingWindow -Name "Backup Staging A"
```

```
PS C:\> $SvcWindowSub = Get-SCServicingWindowSubscription -ServicingWindow $SvcWindow
```

```
PS C:\> $SvcWindowSub
```

## 3: Get the servicing window subscription for a specified virtual machine and a specified servicing window.

The first command gets the virtual machine object named VM01 and stores the object in the \$VM variable.

The second command gets the servicing window object named Backup Staging A and stores the object in the \$SvcWindow variable.

The third command gets the servicing window subscription object for the virtual machine stored in \$VM (VM01) and the servicing window stored in \$SvcWindow (Backup Staging A) and stores the object in the \$SvcWindowSub variable.

The last command displays information about the servicing window subscription stored in \$SvcWindowSub to the user.

```
PS C:\> $VM = Get-VM -Name "VM01"
```

```
PS C:\> $SvcWindow = Get-SCServicingWindow -Name "Backup Staging A"
```

```
PS C:\> $SvcWindowSub = Get-SCServicingWindowSubscription -VM $VM -ServicingWindow $SvcWindow
```

```
PS C:\> $SvcWindowSub
```

## Related topics

[Add-SCServicingWindowSubscription](#)

[Remove-SCServicingWindowSubscription](#)

---

# Get-SCSharedResource

---

## Get-SCSharedResource

Gets resources that are shared with a self-service user or a self-service user role.

### Syntax

Parameter Set: Default

```
Get-SCSharedResource [-UserName <NTAccount> ] [-UserRole <UserRole> ] [-VMMServer  
<ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCSharedResource gets resources that are shared with a self-service user or a self-service user role.

### Parameters

#### **-UserName<NTAccount>**

Specifies a the name of a user. Enter a user name with the format Domain\User.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

#### **-UserRole<UserRole>**

Specifies a user role object.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **SharedResource**

## **Examples**

### **1: Get all resources that are shared with a specific self-service user.**

This command returns all resources that are shared to the user named Katarina.

```
PS C:\> Get-SCSharedResource -UserName "Contoso\Katarina"
```

---

## 2: Get all resources that are shared with a specific self-service user role.

The first command gets the user role object named ContosoSelfServiceUsers and stores the object in the \$Role variable.

The second command returns the resources that are shared to the user role stored in \$Role (ContosoSelfServiceUsers).

```
PS C:\> $Role = Get-SCUserRole -Name "ContosoSelfServiceUsers"
```

```
PS C:\> Get-SCSharedResource -UserRole $Role
```

## 3: Get all resources that are shared with a specific user in a specific user role.

The first command gets the user role object named ContosoSelfServiceUsers and stores the object in the \$Role variable.

The second command returns the resources that are shared with the user named Katarina in the user role stored in \$Role (ContosoSelfServiceUsers).

```
PS C:\> $Role = Get-SCUserRole -Name "ContosoSelfServiceUsers"
```

```
PS C:\> Get-SCSharedResource -UserName "Contoso\Katarina" -UserRole $Role
```

## Related topics

[Grant-SCResource](#)

[Revoke-SCResource](#)

---

# Get-SCSQLDeployment

---

## Get-SCSQLDeployment

Gets a SQL Server deployment.

### Syntax

Parameter Set: ID

```
Get-SCSQLDeployment -ID <Guid> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: SQLConfigurationFile

```
Get-SCSQLDeployment [-SQLConfigurationFile <Script> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: SQLProfile

```
Get-SCSQLDeployment -SQLProfile <SQLProfile> [-Name <String> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

## Detailed Description

The Get-SCSQLDeployment cmdlet gets a SQL Server deployment.

## Parameters

### -ID<Guid>

Specifies the numerical identifier (as a globally unique identifier, or GUID) for a specific object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-SQLConfigurationFile<Script>**

Specifies a SQL Server configuration file.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-SQLProfile<SQLProfile>**

Specifies a SQL Server profile object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

---

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

### **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **SQLDeployment**

### **Examples**

#### **1: Get a specific SQL Server deployment object from an existing SQL Server profile**

The first command gets the SQL Server profile object named SQLProfile01 and stores the object in the \$SQLProfile variable.

The second command gets the SQL Server deployment object named SQL Deployment 01 from the SQL Profile stored in \$SQLProfile, and then stores the object in the \$SQLDeployment variable.

The third command displays the details of the SQL Server deployment stored in \$SQLDeployment to the user.

```
PS C:\> $SQLProfile = Get-SCSQLProfile -Name "SQLProfile01"
```

```
PS C:\> $SQLDeployment = Get-SCSQLDeployment -SQLProfile $SQLProfile -Name "SQL Deployment 01"
```

```
PS C:\> $SQLDeployment
```

### **Related topics**

[Add-SCSQLDeployment](#)

---

[Remove-SCSQLDeployment](#)

[Set-SCSQLDeployment](#)



---

# Get-SCSQLProfile

---

## Get-SCSQLProfile

Gets a SQL Server profile.

### Syntax

Parameter Set: GetAll

```
Get-SCSQLProfile [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: All

```
Get-SCSQLProfile [-All] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: GetByName

```
Get-SCSQLProfile -Name <String> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: GetByVMTemplate

```
Get-SCSQLProfile -VMTemplate <Template> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ID

```
Get-SCSQLProfile [-ID <Guid> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

## Detailed Description

The Get-SCSQLProfile cmdlets gets a SQL Server profile.

## Parameters

### -All

Retrieves a full list of all subordinate objects independent of the parent object. For example, the command Get-SCVirtualDiskDrive -All retrieves all virtual disk drive objects regardless of the virtual machine object or template object that each virtual disk drive object is associated with.

Aliases	none
Required?	false
Position?	named

---

Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-ID<Guid>**

Specifies the numerical identifier (as a globally unique identifier, or GUID) for a specific object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **-VMTemplate<Template>**

Specifies a VMM template object used to create virtual machines.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **SQLProfile**

## **Examples**

### **1: Get a SQL Server profile.**

The first command gets the SQL Server profile object named SQLProfile01 and stores the object in the \$SQLProfile variable.

The second command displays information about the SQL Server profile object stored in \$SQLProfile to the user

```
PS C:\> $SQLProfile = Get-SCSQLProfile -Name "SQLProfile01"
```

```
PS C:\> $SQLProfile
```

---

## Related topics

[New-SCSQLProfile](#)

[Remove-SCSQLProfile](#)

[Set-SCSQLProfile](#)

---

# Get-SCSQLScriptCommand

---

## Get-SCSQLScriptCommand

Gets a SQL Server script for an application deployment.

### Syntax

Parameter Set: Default

```
Get-SCSQLScriptCommand -ApplicationDeployment <ApplicationDeployment> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCSQLScriptCommand cmdlet gets a SQL Server script for an application deployment.

### Parameters

#### **-ApplicationDeployment<ApplicationDeployment>**

Specifies an application deployment object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

#### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
---------	------

---

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **SQLScriptCommand**

## Examples

### 1: Get a specified SQL Server script command for an application deployment.

The first command gets the application profile object named SvcWebAppProfile01 and stores the object in the \$AppProfile variable.

The second command gets the application deployment object named SQLDataTierApp01 for the application profile stored in \$AppProfile, and then stores the object in the \$AppDeployment variable.

The last command gets the first PreInstall SQL Server script (deployment order 1, SQL script type PreInstall) associated with the application deployment stored in \$AppDeployment.

```
PS C:\> $AppProfile = Get-SCApplicationProfile -Name "SvcWebAppProfile01"
```

```
PS C:\> $AppDeployment = Get-SCApplicationDeployment -ApplicationProfile $AppProfile -Name "SQLDataTierApp01"
```

```
PS C:\> Get-SCSQLScriptCommand -ApplicationDeployment $AppDeployment | where  
{$_ .DeploymentOrder -eq "1" -and $_.SQLScriptType -eq "PreInstall"}
```

## Related topics

[Add-SCSQLScriptCommand](#)

[Remove-SCSQLScriptCommand](#)

[Set-SCSQLScriptCommand](#)

---

# Get-SCSSASConnection

---

## Get-SCSSASConnection

Gets the existing SQL Server Analysis Services connection.

### Syntax

Parameter Set: Default

```
Get-SCSSASConnection [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCSSASConnection cmdlet gets the existing SQL Server Analysis Services (SSAS) connection.

### Parameters

#### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

#### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

---

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **SQLAnalysisServicesServer**

## Examples

### 1: Get the existing SQL Server Analysis Services connection.

This command gets the existing SSAS connection and displays information about the connection to the user.

```
PS C:\> Get-SCSSASConnection
```

## Related topics

[New-SCSSASConnection](#)

[Remove-SCSSASConnection](#)



---

# Get-SCSSHKey

---

## Get-SCSSHKey

Gets Linux Administrator Ssh Key objects from the VMM library.

### Syntax

Parameter Set: All

```
Get-SCSSHKey [-All] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: EquivalentResourceParamSet

```
Get-SCSSHKey -FamilyName <String> [-Release <String> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ID

```
Get-SCSSHKey [-ID <Guid> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: NameParamSet

```
Get-SCSSHKey -Name <String> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCSSHKey cmdlet gets Linux Administrator SSH Key objects from the VMM library.

### Parameters

#### -All

Retrieves a full list of all subordinate objects independent of the parent object. For example, the command `Get-SCVirtualDiskDrive -All` retrieves all virtual disk drive objects regardless of the virtual machine object or template object that each virtual disk drive object is associated with.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false

---

Accept Wildcard Characters?	false
-----------------------------	-------

### **-FamilyName<String>**

Specifies a family name for a physical resource in the VMM library.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-ID<Guid>**

Specifies the numerical identifier (as a globally unique identifier, or GUID) for a specific object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false

Accept Wildcard Characters?	false
-----------------------------	-------

### **-Release<String>**

Specifies a string that describes the release of a library resource. VMM automatically creates a release value for every resource imported into the library. After the resource has been imported, the string can be customized.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

### **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **LinuxAdministratorSshKey**

---

## Examples

### 1: Get all SSHKey objects in the VMM library.

This command gets all SSHKey objects in the VMM library and displays information about each to the user.

```
PS C:\> $SSHKey = Get-SCSSHKey -All
```

### 2: Get an SSHKey by its name.

The first command gets the SSHKey object named My.sshkey and stores the object in the \$SSHKey variable.

The second command displays information about the SSHKey object stored in \$SSHKey to the user

```
PS C:\> $SSHKey = Get-SCSSHKey -Name "My.sshkey"
```

```
PS C:\> $SSHKey
```

### 3: Get all SSHKey objects with the same FamilyName value.

This command gets all SSHKey objects from the VMM library with the FamilyName value of Family01, and displays information about each SSH key to the user.

```
PS C:\> Get-SCSSHKey -FamilyName "Family01"
```

## Related topics

[Remove-SCSSHKey](#)

[Set-SCSSHKey](#)

---

# Get-SCStaticIPAddressPool

---

## Get-SCStaticIPAddressPool

Gets a static IP address pool.

### Syntax

Parameter Set: All

```
Get-SCStaticIPAddressPool [[-Name] <String> ] [-IPAddress <String> ] [-IPv4] [-IPv6] [-IsMulticast] [-Subnet <String> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ByCloud

```
Get-SCStaticIPAddressPool [[-Name] <String> ] -Cloud <Cloud> [-IPAddress <String> ] [-IsMulticast] [-Subnet <String> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ByHostGroup

```
Get-SCStaticIPAddressPool [[-Name] <String> ] -VMHostGroup <HostGroup> [-IPAddress <String> ] [-IPv4] [-IPv6] [-IsMulticast] [-Subnet <String> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ById

```
Get-SCStaticIPAddressPool [[-Name] <String> ] -ID <Guid> [-IPAddress <String> ] [-IsMulticast] [-Subnet <String> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ByLogicalNetworkDefinition

```
Get-SCStaticIPAddressPool [[-Name] <String> ] -LogicalNetworkDefinition <LogicalNetworkDefinition> [-IPAddress <String> ] [-IPv4] [-IPv6] [-IsMulticast] [-Subnet <String> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ByVMSubnet

```
Get-SCStaticIPAddressPool [[-Name] <String> ] -VMSubnet <VMSubnet> [-IPAddress <String> ] [-IPv4] [-IPv6] [-IsMulticast] [-Subnet <String> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCStaticIPAddressPool cmdlet gets one or more Virtual Machine Manager (VMM) static IP address pools.

---

## Parameters

### **-Cloud<Cloud>**

Specifies a private cloud object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-ID<Guid>**

Specifies the numerical identifier (as a globally unique identifier, or GUID) for a specific object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-IPAddress<String>**

Specifies an IPv4 or IPv6 address.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-IPv4**

Indicates if an IPv4 address is needed

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-IPv6**

Indicates if an IPv6 address is needed

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-IsMulticast**

Indicates that the IP address is a multicast address or that the IP address pool contains a multicast IP address range.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false

Accept Wildcard Characters?	false
-----------------------------	-------

### **-LogicalNetworkDefinition<LogicalNetworkDefinition>**

Specifies a logical network definition (also called a network site) that contains the subnet that the IP address pool serves as specified by the Subnet parameter.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	false
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Subnet<String>**

Specifies an IP subnet (IPv4 or IPv6) in Classless Inter-Domain Routing (CIDR) notation.

Example format for an IPv4 subnet: 192.168.0.1/24

Example format for an IPv6 subnet: FD4A:29CD:184F:3A2C::/64

NOTE: An IP subnet cannot overlap with any other subnet in a host group or child host groups.

Aliases	none
Required?	false



Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMHostGroup<HostGroup>**

Specifies a virtual machine host group object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-VMSubnet<VMSubnet>**

Specifies a VM subnet object.

To obtain a VM subnet object, use the Get-SCVMSubnet cmdlet.

Aliases	none
---------	------

---

Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **StaticIPAddressPool**

## Examples

### 1: Get all available IPv4 IP address pools for a specified subnet.

This command gets the static IP address pool for the specified IPv4 subnet address.

```
PS C:\> Get-SCStaticIPAddressPool -IPv4 -Subnet "10.0.0.0/24"
```

### 2: Get all IPv4 IP address pools for a specified host group.

The first command gets the host group with the path of All Hosts\HostGroup02\Production and stores it in the \$HostGroup variable.

The second command gets the static IPv4 IP address pools for the host group stored in \$HostGroup.

```
PS C:\> $HostGroup = Get-SCVMHostGroup | where { $_.Path -eq "All  
Hosts\HostGroup02\Production" }
```

```
PS C:\> Get-SCStaticIPAddressPool -IPv4 -VMHostGroup $HostGroup
```

## Related topics

[New-SCStaticIPAddressPool](#)

[Remove-SCStaticIPAddressPool](#)

[Set-SCStaticIPAddressPool](#)

---

# Get-SCStep

---

## Get-SCStep

Gets the steps for the specified VMM job.

### Syntax

Parameter Set: Default

```
Get-SCStep [-Job] <Task> [-Name <String> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCStep cmdlet gets the steps for the specified Virtual Machine Manager (VMM) job.

A job is composed of one or more steps, each of which has its own status. An earlier step must complete or be skipped before the next step runs.

### Parameters

#### **-Job<Task>**

Specifies a VMM job object.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

#### **-Name<String>**

Specifies the name of a VMM object.

---

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **Step**

## Examples

### 1: Get all steps for a specified job.

The first command gets the VMM job object with the ID cb3a0f0a-9fbc-4bd0-a999-3fae8cd77177 and stores the object in the \$Job variable.

The second command gets the step object for the VMM job in \$Job.

The last command lists information about each step for the VMM job, including a description of the step, its progress, and its status.

```
PS C:\> $Job = Get-SCJob -ID "cb3a0f0a-9fbc-4bd0-a999-3fae8cd77177"
```

```
PS C:\> $Steps = Get-SCStep -Job $Job
```

```
PS C:\> $Steps.Children
```

## Related topics

[Get-SCJob](#)

---

# Get-SCStorageArray

---

## Get-SCStorageArray

Gets a storage array object.

### Syntax

Parameter Set: All

```
Get-SCStorageArray [[-Name] <String> ] [-All] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: GetConnectableArray

```
Get-SCStorageArray [[-Name] <String> ] -VMHost <Host[]> [-ConnectedToAllHost] [-FibreChannelOnly] [-iSCSIOnly] [-SASOnly] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ID

```
Get-SCStorageArray [[-Name] <String> ] [-ID <Guid> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCStorageArray cmdlet gets a storage array object from the Virtual Machine Manager (VMM) database.

You must install the storage provider on an available computer prior to discovering the storage resources.

For more information about configuring storage, see [Configuring Storage Overview](http://go.microsoft.com/fwlink/?LinkID=212013) in the Microsoft TechNet library at <http://go.microsoft.com/fwlink/?LinkID=212013>.

### Parameters

#### -All

Retrieves a full list of all subordinate objects independent of the parent object. For example, the command `Get-SCVirtualDiskDrive -All` retrieves all virtual disk drive objects regardless of the virtual machine object or template object that each virtual disk drive object is associated with.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-ConnectedToAllHost**

Indicates that a storage array is connected to all hosts.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-FibreChannelOnly**

Indicates that only Fibre Channel arrays are returned.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-ID<Guid>**

Specifies the numerical identifier (as a globally unique identifier, or GUID) for a specific object.

Aliases	none
---------	------

---

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-iSCSIOnly**

Indicates that only iSCSI arrays are returned.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	false
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-SASOnly**

Indicates that only SAS storage arrays are returned.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMHost<Host[]>**

Specifies a virtual machine host object. VMM supports Hyper-V hosts, VMware ESX hosts, and Citrix XenServer hosts.

For more information about each type of host, type: "Get-Help Add-SCVMHost -detailed". See the examples for a specific cmdlet to determine how that cmdlet uses this parameter.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false



---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **StorageArray**

## Examples

### 1: Get a storage array by its name.

This command gets the storage array named SANArray.

```
PS C:\> Get-SCStorageArray -Name "SANArray"
```

## Related topics

[Set-SCStorageArray](#)

---

# Get-SCStorageClassification

---

## Get-SCStorageClassification

Gets a storage classification object.

### Syntax

Parameter Set: All

```
Get-SCStorageClassification [[-Name] <String> ] [-All] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ID

```
Get-SCStorageClassification [[-Name] <String> ] [-ID <Guid> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCStorageClassification cmdlet gets a storage classification object.

### Parameters

#### -All

Retrieves a full list of all subordinate objects independent of the parent object. For example, the command Get-SCVirtualDiskDrive -All retrieves all virtual disk drive objects regardless of the virtual machine object or template object that each virtual disk drive object is associated with.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-ID<Guid>**

Specifies the numerical identifier (as a globally unique identifier, or GUID) for a specific object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	false
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **StorageClassification**

## Examples

### 1: Get a storage classification by its name.

This command gets the storage classification named StorageClassification01.

```
PS C:\> Get-SCStorageClassification -Name "StorageClassification01"
```

## Related topics

[New-SCStorageClassification](#)

[Remove-SCStorageClassification](#)

[Set-SCStorageClassification](#)

---

# Get-SCStorageDisk

---

## Get-SCStorageDisk

Gets a storage disk object for the specified host from the VMM database.

### Syntax

Parameter Set: NoFilter

```
Get-SCStorageDisk [[-Name] <String> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: FilterByVMHost

```
Get-SCStorageDisk [[-Name] <String> ] -VMHost <Host> [ <CommonParameters>]
```

### Detailed Description

The Get-SCStorageDisk cmdlet gets one or more storage disk objects for the specified host from the Virtual Machine Manager (VMM) database. You can use this cmdlet with the New-SCVirtualDiskDrive cmdlet to attach a pass-through disk on a virtual machine to a physical hard disk on the host on which that virtual machine is deployed.

### Parameters

#### **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	false
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-VMHost<Host>**

Specifies a virtual machine host object. VMM supports Hyper-V hosts, VMware ESX hosts, and Citrix XenServer hosts.

For more information about each type of host, type: "Get-Help Add-SCVMHost -detailed". See the examples for a specific cmdlet to determine how that cmdlet uses this parameter.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **StorageDisk**

---

## Examples

### 1: Get all hard disk drives on the specified host.

The first command gets the host object named VMHost01 and stores the object in the \$VMHost variable.

The second command gets all hard disk drive objects from the host stored in \$VMHost and displays information about those objects to the user.

```
PS C:\> $VMHost = Get-SCVMHost -ComputerName "VMHost01.Contoso.com"
PS C:\> Get-SCStorageDisk -VMHost $VMHost
```

### 2. Get a specific hard disk drive on the host by name.

The first command gets the host object named VMHost02 and uses the pipeline operator to pass VMHost02 to the Get-SCStorageDisk cmdlet which gets the hard disk drive object named "PhysicalDrive0" for the host. The command then stores the hard disk drive object in the \$StorageDisk variable.

The second command displays the contents of \$StorageDisk to the user.

```
PS C:\> $StorageDisk = Get-SCVMHost -ComputerName "VMHost02.Contoso.com" | Get-SCStorageDisk
-Name "\\.\PhysicalDrive0"
PS C:\> $StorageDisk
```

## Related topics

[New-SCVirtualDiskDrive](#)

[Mount-SCStorageDisk](#)

---

# Get-SCStorageFileServer

---

## Get-SCStorageFileServer

Gets a storage file server.

### Syntax

Parameter Set: All

```
Get-SCStorageFileServer [[-Name] <String> ] [-All] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ID

```
Get-SCStorageFileServer [[-Name] <String> ] [-ID <Guid> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCStorageFileServer cmdlets gets a storage file server.

### Parameters

#### -All

Retrieves a full list of all subordinate objects independent of the parent object. For example, the command Get-SCVirtualDiskDrive -All retrieves all virtual disk drive objects regardless of the virtual machine object or template object that each virtual disk drive object is associated with.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false



---

## **-ID<Guid>**

Specifies the numerical identifier (as a globally unique identifier, or GUID) for a specific object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	false
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Examples

### 1: Get a specific storage file server.

The first command gets the file server object named FileServer01 and stores the object in the \$FileServer variable.

The second command displays information about the file server stored in \$FileServer to the user.

```
PS C:\> $FileServer = Get-SCStorageFileServer -Name "FileServer01.Contoso.com"
```

```
PS C:\> $FileServer
```

## Related topics

[Set-SCStorageFileServer](#)

---

# Get-SCStorageFileShare

---

## Get-SCStorageFileShare

Gets a storage file share.

### Syntax

Parameter Set: All

```
Get-SCStorageFileShare [[-Name] <String> ] [-All] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: FilterByLibraryServer

```
Get-SCStorageFileShare [[-Name] <String> ] -LibraryServer <LibraryServer> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: FilterByVMHost

```
Get-SCStorageFileShare [[-Name] <String> ] -VMHost <Host> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: FilterByVMHostCluster

```
Get-SCStorageFileShare [[-Name] <String> ] -VMHostCluster <HostCluster> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ID

```
Get-SCStorageFileShare [[-Name] <String> ] [-ID <Guid> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

## Detailed Description

The Get-SCStorageFileShare cmdlet gets a storage file share from the Virtual Machine Manager (VMM) database.

## Parameters

### -All

Retrieves a full list of all subordinate objects independent of the parent object. For example, the command Get-SCVirtualDiskDrive -All retrieves all virtual disk drive objects regardless of the virtual machine object or template object that each virtual disk drive object is associated with.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-ID<Guid>**

Specifies the numerical identifier (as a globally unique identifier, or GUID) for a specific object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-LibraryServer<LibraryServer>**

Specifies a VMM library server object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	false
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMHost<Host>**

Specifies a virtual machine host object. VMM supports Hyper-V hosts, VMware ESX hosts, and Citrix XenServer hosts.

For more information about each type of host, type: "Get-Help Add-SCVMHost -detailed". See the examples for a specific cmdlet to determine how that cmdlet uses this parameter.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-VMHostCluster<HostCluster>**

Specifies a VMM host cluster object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

---

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

### **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **StorageFileShare**

### **Examples**

#### **1: Get a storage file share by its name.**

The first command gets the storage file share object named FileShare01 and stores the object in the \$FileShare variable.

The second command displays information about the storage file share stored in FileShare01 to the user.

```
PS C:\> $FileShare = Get-SCStorageFileShare -Name "FileShare01"
```

```
PS C:\> $FileShare
```

### **Related topics**

[New-SCStorageFileShare](#)

[Register-SCStorageFileShare](#)

[Remove-SCStorageFileShare](#)

[Repair-SCStorageFileShare](#)

[Set-SCStorageFileShare](#)

---

[Unregister-SCStorageFileShare](#)

---

# Get-SCStorageLogicalUnit

---

## Get-SCStorageLogicalUnit

Gets a storage logical unit object.

### Syntax

Parameter Set: All

```
Get-SCStorageLogicalUnit [[-Name] <String> ] [-All] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ID

```
Get-SCStorageLogicalUnit [[-Name] <String> ] [-ID <Guid> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCStorageLogicalUnit gets a storage logical unit object from the Virtual Machine Manager (VMM) database.

### Parameters

#### -All

Retrieves a full list of all subordinate objects independent of the parent object. For example, the command Get-SCVirtualDiskDrive -All retrieves all virtual disk drive objects regardless of the virtual machine object or template object that each virtual disk drive object is associated with.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false



---

## **-ID<Guid>**

Specifies the numerical identifier (as a globally unique identifier, or GUID) for a specific object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	false
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **StorageLogicalUnit**

## Examples

### 1: Get a storage logical unit by its name.

This command gets the storage logical unit object named LUN01 and displays information about the storage logical unit to the user.

```
PS C:\> Get-SCStorageLogicalUnit -Name "LUN01"
```

## Related topics

[New-SCStorageLogicalUnit](#)

[Register-SCStorageLogicalUnit](#)

[Remove-SCStorageLogicalUnit](#)

[Set-SCStorageLogicalUnit](#)

[Unregister-SCStorageLogicalUnit](#)

---

# Get-SCStoragePool

---

## Get-SCStoragePool

Gets a storage pool object from the VMM database.

### Syntax

Parameter Set: All

```
Get-SCStoragePool [[-Name] <String> ] [-All] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ID

```
Get-SCStoragePool [[-Name] <String> ] [-ID <Guid> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCStoragePool cmdlet gets one or more storage pool objects from the Virtual Machine Manager (VMM) database.

### Parameters

#### -All

Retrieves a full list of all subordinate objects independent of the parent object. For example, the command Get-SCVirtualDiskDrive -All retrieves all virtual disk drive objects regardless of the virtual machine object or template object that each virtual disk drive object is associated with.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-ID<Guid>**

Specifies the numerical identifier (as a globally unique identifier, or GUID) for a specific object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	false
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **StoragePool**

## Examples

### 1: Get a storage pool by its name.

This command gets the storage pool named StoragePool01.

```
PS C:\> Get-SCStoragePool -Name "StoragePool01"
```

## Related topics

[Set-SCStoragePool](#)

---

# Get-SCStorageProvider

---

## Get-SCStorageProvider

Gets a storage provider object.

### Syntax

Parameter Set: All

```
Get-SCStorageProvider [[-Name] <String> ] [-All] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ID

```
Get-SCStorageProvider [[-Name] <String> ] [-ID <Guid> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Get-StorageProvider cmdlet gets one or more storage provider objects.

### Parameters

#### -All

Retrieves a full list of all subordinate objects independent of the parent object. For example, the command Get-SCVirtualDiskDrive -All retrieves all virtual disk drive objects regardless of the virtual machine object or template object that each virtual disk drive object is associated with.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-ID<Guid>**

Specifies the numerical identifier (as a globally unique identifier, or GUID) for a specific object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	false
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **StorageProvider**

## Examples

### 1: Get a storage provider by its name.

This command gets the storage provider named StorProv01 and stores it in the \$Provider variable.

```
PS C:\> $Provider = Get-SCStorageProvider -Name "StorProv01.Contoso.com"
```

## Related topics

[Add-SCStorageProvider](#)

[Read-SCStorageProvider](#)

[Remove-SCStorageProvider](#)

[Set-SCStorageProvider](#)



---

# Get-SCStorageVolume

---

## Get-SCStorageVolume

Gets a storage volume object from a host managed by VMM.

### Syntax

Parameter Set: NoFilter

```
Get-SCStorageVolume [[-Name] <String> ] [-VMMServer <ServerConnection> ] [ <CommonParameters> ]
```

Parameter Set: FilterByVMHost

```
Get-SCStorageVolume [[-Name] <String> ] -VMHost <Host> [ <CommonParameters> ]
```

### Detailed Description

The Get-SCStorageVolume cmdlet gets one or more storage volume objects from a host managed by Virtual Machine Manager (VMM).

The information returned includes, but is not limited to, the following:

- Name - The name of each host volume (such as C:\, D:\, E:\).
- StorageVolumeID - The volume ID (a GUID) for each host volume.

The host volume ID is unique across your VMM environment.

- MountPoints - The mount points for each volume.

A single volume, such as C:\, can contain multiple mount points.

- Capacity - The storage capacity of each volume.
- FreeSpace - The amount of free space on each volume.
- VolumeLabel - A user-defined label for this volume (if any).
- IsSANMigrationPossible - A flag indicating whether or not SAN migration is available.
- IsClustered - A flag indicating whether the volume is local storage or shared storage (that is, uses external storage, such as SAN or iSCSI) and a clustered disk resource exists for this volume.
- InUse - A flag that is set to True when one of the highly available virtual machines managed by VMM is using this volume.
- VMHost - The FQDN name of the host on which each volume resides.
- IsAvailableForPlacement - A flag indicating whether this volume is available as a location on which to deploy virtual machines

---

on this host.

- ServerConnection - The VMM server connection that is managing the host that this volume belongs to.
- ID - The ID (a GUID) for each volume.

## Parameters

### **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	false
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMHost<Host>**

Specifies a virtual machine host object. VMM supports Hyper-V hosts, VMware ESX hosts, and Citrix XenServer hosts.

For more information about each type of host, type: "Get-Help Add-SCVMHost -detailed". See the examples for a specific cmdlet to determine how that cmdlet uses this parameter.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

---

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **StorageVolume**

## Examples

### 1: Get all volumes on the specified host server.

The first command gets the host object named VMHost01 and stores the object in the \$VMHost variable.

The second command gets all drive volume objects from VMHost01 and displays information about these volumes to the user.

NOTE: To translate the capacity and free space from bytes into larger units of measure, divide the number of bytes by 1024 to get kilobytes (KB); divide the result by 1024 to get megabytes (MB); and divide that result by 1024 to get gigabytes (GB).

```
PS C:\> $VMHost = Get-SCVMHost -ComputerName "VMHost01.Contoso.com"
```

```
PS C:\> Get-SCStorageVolume -VMHost $VMHost
```

### 2: Get the specified volume on a host.

The first command gets the host object named VMHost02 and stores the object in the \$VMHost variable.

The second command gets the drive volume named C:\ from VMHost02 and displays information about this volume to the user.

```
PS C:\> $VMHost = Get-SCVMHost -ComputerName "VMHost02.Contoso.com"
```

```
PS C:\> Get-SCStorageVolume -VMHost $VMHost -Name "C:\"
```

---

### 3: Get all volumes on VMware ESX hosts that contain the string "SharedStorage" in the volume name.

The first command gets all host objects from VMMServer01, selects only those host objects whose virtualization platform is VMware ESX, and then stores those host objects in \$VMHost.

NOTE: This example assumes that the names of all volumes on these ESX Servers include the string "storage", but that only some of those volumes' names include the string "SharedStorage."

The second command passes each ESX host object in \$VMHost to the Get-SCVMStorageVolume cmdlet, which gets the volume objects on these hosts and then, in turn, passes the volume objects to "select" (the alias for the Select-Object cmdlet). The Select-Object cmdlet displays the volume name and the host that volume resides on for those volumes whose name contains the string "SharedStorage".

```
PS C:\> $VMHost = Get-SCVMHost -VMMServer "VMMServer01.Contoso.com" | where {
$_VirtualizationPlatform -eq "VMwareESX" }
```

```
PS C:\> $VMHost | Get-SCStorageVolume | select -property Name, VMHost | where { $_.Name -
match "SharedStorage" }
```

### Related topics

[Get-SCVMHost](#)

[Set-SCStorageVolume](#)

---

# Get-SCTag

---

## Get-SCTag

Gets existing tags.

## Syntax

Parameter Set: Default

```
Get-SCTag [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

## Detailed Description

The Get-SCTag cmdlet gets all existing user-defined tags.

## Parameters

### -VMMServer<ServerConnection>

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- 
- `String[]`

## Examples

### 1: Retrieve all tags on a specified VMM management server.

This command gets all tags on VMMServer01.

```
PS C:\> Get-SCTag -VMMServer "VMMServer01"
```

## Related topics

[Get-SCDriverPackage](#)

[Set-SCDriverPackage](#)

---

# Get-SCTemplatePackage

---

## Get-SCTemplatePackage

Gets an exported service template or virtual machine template package at a specified location.

### Syntax

Parameter Set: Default

```
Get-SCTemplatePackage -Path <String> [ <CommonParameters>]
```

### Detailed Description

The Get-SCTemplatePackage gets an exported service template or virtual machine template package at a specified location. After you get the template package object, you can read the properties of the object, or you can input the object into the Import-SCTemplate cmdlet.

For more information about importing a template package, type: "Get-Help Import-SCTemplate -detailed".

### Parameters

#### **-Path<String>**

Specifies the destination path for the operation.

Example formats:

Local path -Path "F:\"

UNC path -Path "\\Library\Templates"

Volume GUID path -Path "\\?\Volume{4703c1ea-8ae7-11db-b473-00123f7603e3}\"

VMware ESX path -Path "[storage1]\MyVMwareFolderForVMs\MyVM.vmx"

Citrix XenServer path - Path "Local storage[99b6212f-b63d-c676-25f9-d6c460992de7]"

Wildcards are supported for "Get" cmdlets and when you specify the UNC path:

Example format:

UNC path -Path "\\VMHostServer\MyVMs\\*VM\*"

Aliases	none
Required?	true

---

Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **TemplatePackage[]**
- **TemplatePackage**

## Examples

### 1: Get a template package from a specified location.

This command gets the template package object in C:\TemplateExports and stores the object in the \$TemplatePackage variable.

```
PS C:\> $TemplatePackage = Get-SCTemplatePackage -Path "C:\TemplateExports"
```

## Related topics

[Export-SCTemplate](#)

[Import-SCTemplate](#)



---

# Get-SCUpdate

---

## Get-SCUpdate

Gets one or more updates.

### Syntax

Parameter Set: ID

```
Get-SCUpdate [-ID <Guid> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: BulletinId

```
Get-SCUpdate [-SecurityBulletinId <String> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: KBArticle

```
Get-SCUpdate [-KBArticle <String> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: Name

```
Get-SCUpdate [-Name <String> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: Newest

```
Get-SCUpdate [-Newest <Int32> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

## Detailed Description

The Get-SCUpdate cmdlet gets one or more updates. An update contains metadata that enables the ability to determine the applicability and installation status of an update that is assigned to a target computer using a baseline.

## Parameters

### -ID<Guid>

Specifies the numerical identifier (as a globally unique identifier, or GUID) for a specific object.

Aliases	none
Required?	false
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-KBArticle<String>**

Specifies the KB article ID number for an update.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-Newest<Int32>**

Returns all jobs created in the last specified number of hours, or returns the specified number of most recent software updates.

Example format to return all jobs created in the last 48 hours: `Get-SCJob -Newest 48`

Example format to return the 10 newest updates: `Get-SCUpdate -Newest 10`

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-SecurityBulletinId<String>**

Specifies the Microsoft Security Response Center (MSRC) Security Bulletin ID for an update.

Example format: -SecurityBulletinId "MS05-045"

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- Update

## Examples

### 1: Get a specific update by its bulletin identification number.

This command gets the security bulletin update named MS05-51.

```
PS C:\> Get-SCUpdate -SecurityBulletinID "MS05-051"
```

### 2: Get a specific update by its KBArticle number.

This command gets the updated identified by KBArticle 93051.

```
PS C:\> Get-SCUpdate -KBArticle "93051"
```

## Related topics

[Set-SCUpdate](#)

---

# Get-SCUpdateServer

---

## Get-SCUpdateServer

Gets the Windows Server Update Services computer that has been added to VMM.

### Syntax

Parameter Set: ID

```
Get-SCUpdateServer [-ID <Guid> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: Name

```
Get-SCUpdateServer [-ComputerName <String> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCUpdateServer cmdlet gets the Microsoft Windows Server Update Services (WSUS) computer that has been added to Virtual Machine Manager (VMM).

For more information about adding an update server to VMM, type: "Get-Help Add-SCUpdateServer".

### Parameters

#### **-ComputerName<String>**

Specifies the name of a computer that VMM can uniquely identify on your network. Valid formats are: FQDN, IPv4 or IPv6 address, or NetBIOS name.

NOTE: See the examples for a specific cmdlet to determine how that cmdlet specifies the computer name.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-ID<Guid>**

Specifies the numerical identifier (as a globally unique identifier, or GUID) for a specific object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **UpdateServer**

---

## Examples

### 1: Get the update server.

This command gets the update server object named WSUSComputer01 and stores the object in the \$UpdateServer variable.

```
PS C:\> $UpdateServer = Get-SCUpdateServer -ComputerName "WSUSComputer01"
```

## Related topics

[Add-SCUpdateServer](#)

[Remove-SCUpdateServer](#)

[Set-SCUpdateServer](#)

---

# Get-SCUplinkPortProfileSet

---

## Get-SCUplinkPortProfileSet

Gets an uplink port profile set.

### Syntax

Parameter Set: All

```
Get-SCUplinkPortProfileSet [[-Name] <String> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ById

```
Get-SCUplinkPortProfileSet [[-Name] <String> ] -ID <Guid> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: LogicalSwitch

```
Get-SCUplinkPortProfileSet [[-Name] <String> ] -LogicalSwitch <LogicalSwitch> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

## Detailed Description

The Get-SCUplinkPortProfileSet cmdlet gets uplink port profile sets. You can get all uplink port profile sets, an uplink port profile set by its name or ID, or all uplink port profile sets for a particular logical switch.

## Parameters

### -ID<Guid>

Specifies the numerical identifier (as a globally unique identifier, or GUID) for a specific object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false



---

## **-LogicalSwitch<LogicalSwitch>**

Specifies a logical switch object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	false
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## Examples

### 1: Get an uplink port profile set by its name.

This command gets the uplink port profile set named UplinkPortProfileSet01.

```
PS C:\> Get-SCUplinkPortProfileSet -Name "UplinkPortProfileSet01"
```

### 2: Get all uplink port profile sets for a logical switch.

The first command gets the logical switch object named LogicalSwitch01 and stores the object in the \$LogSwitch variable.

The second command gets all uplink port profile sets for LogicalSwitch01 and displays them for the user.

```
PS C:\> $LogSwitch = Get-SCLogicalSwitch -Name "LogicalSwitch01"
```

```
PS C:\> Get-SCUplinkPortProfileSet -LogicalSwitch $LogSwitch
```

## Related topics

[New-SCUplinkPortProfileSet](#)

[Remove-SCUplinkPortProfileSet](#)

[Set-SCUplinkPortProfileSet](#)

---

# Get-SCUserRole

---

## Get-SCUserRole

Gets a VMM user role.

### Syntax

Parameter Set: Default

```
Get-SCUserRole [[-Name] <String> ] [-ID <Guid> ] [-UserRoleProfile {Administrator | DelegatedAdmin | TenantAdmin | SelfServiceUser | ReadOnlyAdmin} ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCUserRole cmdlet gets one or more Virtual Machine Manager (VMM) user roles. VMM uses role-based security to define the boundaries within which members of a given user role can operate and the set of allowed operations members of a user role can perform.

For information about creating user roles, type: "Get-Help New-SCUserRole -detailed".

For information about setting the properties of a user role, including the scope for delegated and read-only administrators and the scope and actions for self-service users, type: "Get-Help Set-SCUserRole -detailed".

### Parameters

#### -ID<Guid>

Specifies the numerical identifier (as a globally unique identifier, or GUID) for a specific object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	false
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-UserRoleProfile<Profile>**

Specifies the type of profile to use as the basis for the user role. Valid values are: DelegatedAdmin, ReadOnlyAdmin, SelfServiceUser.

The acceptable values for this parameter are:

Administrator	
DelegatedAdmin	
TenantAdmin	
SelfServiceUser	
ReadOnlyAdmin	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

---

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **UserRole**

## Examples

### 1: Get all VMM user roles.

This command gets all VMM user role objects on VMMServer01 and displays information about each user role.

```
PS C:\> Get-SCUserRole -VMMServer "VMMServer01.Contoso.com"
```

### 2: Get a specific user role by name.

This command gets the user role object named Administrator, and then displays information about that user role to the user.

```
PS C:\> Get-SCUserRole -Name "Administrator"
```

### 3: Get a specific user role by UserRoleProfile.

This command gets the user role objects on VMMServer01 that have a UserRoleProfile value of "SelfServiceUser", and then displays information about these users role to the user.

```
PS C:\> Get-SCUserRole -VMMServer "VMMServer01.Contoso.com" -UserRoleProfile "SelfServiceUser"
```

---

## 4. Display properties and other information about user role objects.

The first command gets all user role objects on VMMServer01 and stores the objects in the \$UserRoles variable.

The second command passes each user role object in \$UserRoles to "select" (the alias for the Select-Object cmdlet) and then displays the name, user role profile, parent user role, and cloud for each user role

The last command passes each user role in \$UserRoles to the Get-Member cmdlet, which displays the .NET type for each user role and the methods and properties associated with each user role type.

```
PS C:\> $UserRoles = Get-SCUserRole -VMMServer "VMMServer01.Contoso.com"
PS C:\> $UserRoles | select Name, UserRoleProfile, ParentUserRole, Cloud
PS C:\> $UserRoles | Get-Member
```

### Related topics

[Grant-SCResource](#)

[New-SCUserRole](#)

[Remove-SCUserRole](#)

[Revoke-SCResource](#)

[Set-SCUserRole](#)

---

# Get-SCUserRoleMembership

---

## Get-SCUserRoleMembership

Gets information about the user roles of which the current user or a specified user is a member.

### Syntax

Parameter Set: Default

```
Get-SCUserRoleMembership [[-UserName] <String> ] [-ForSharing] [-Resource <ClientObject> ]  
[-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCUserRoleMembership cmdlet gets information about the user roles of which the current user or a specified user is a member.

### Parameters

#### -ForSharing

Returns information about self-service user roles to which the user belongs that can share resources.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

#### -Resource<ClientObject>

Specifies a resource object.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-UserName<String>**

Specifies a the name of a user. Enter a user name with the format Domain\User.

Aliases	none
Required?	false
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).



---

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **UserRoleMembership**

## Examples

### 1: Get information about the user roles the current user is a member of.

This command returns the Name, Description, and UserRoleProfile for each user role the current user is a member of.

```
PS C:\> Get-SCUserRoleMembership
```

## Related topics

[Get-SCUserRole](#)

[New-SCUserRole](#)

[Remove-SCUserRole](#)

[Set-SCUserRole](#)

---

# Get-SCUserRoleQuota

---

## Get-SCUserRoleQuota

Gets a user role quota object.

### Syntax

Parameter Set: Default

```
Get-SCUserRoleQuota [[-UserRole] <UserRole> ] [-Cloud <Cloud> ] [-QuotaPerUser <Boolean> ]  
[-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCUserRoleQuota gets a Virtual Machine Manager (VMM) user role quota object. Two user role quota objects are returned for a self-service user role per private cloud: one object contains information about the role-level quota, and the other object contains information about member-level quota. The QuotaPerUser property allows you to return just one of the objects. When QuotaPerUser is set to True, only the member-level quota object is returned. When set to False, only the role-level quota object is returned.

### Parameters

#### **-Cloud<Cloud>**

Specifies a private cloud object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-QuotaPerUser<Boolean>**

Indicates whether the cmdlet sets or retrieves user level quotas or member level quotas. Specifying \$True indicates member level quotas. Specifying \$False indicates role level quotas. If the parameter is not used, both quotas are set or returned.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-UserRole<UserRole>**

Specifies a user role object.

Aliases	none
Required?	false
Position?	1
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **UserRoleQuota**

## Examples

### 1: Get per-user virtual machine count quota for a specific private cloud.

The first command gets the private cloud object named Cloud01 and stores the object in the \$Cloud variable.

The second command gets the user role object named ContosoSelfServiceUsers and stores the object in the \$Role variable.

The third command gets the user role quota for the private cloud stored in \$Cloud (Cloud01) and user role stored in \$Role (ContosoSelfServiceUsers). The QuotaPerUser parameter set to true indicates that the quota for members will be returned.

The last command displays the virtual machine count quota to the user.

```
PS C:\> $Cloud = Get-SCCloud -Name "Cloud01"
```

```
PS C:\> $Role = Get-SCUserRole -Name "ContosoSelfServiceUsers"
```

```
PS C:\> $Quota = Get-SCUserRoleQuota -Cloud $Cloud -UserRole $Role -QuotaPerUser $True
```

```
PS C:\> Write-Output $Quota.VMCount
```

## Related topics

[Set-SCUserRoleQuota](#)

---

# Get-SCVirtualCOMPort

---

## Get-SCVirtualCOMPort

Gets a VMM virtual communication (COM) port object from a virtual machine, a virtual machine template, or a hardware profile.

### Syntax

Parameter Set: All

```
Get-SCVirtualCOMPort -All [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: HardwareProfile

```
Get-SCVirtualCOMPort -HardwareProfile <HardwareProfile> [ <CommonParameters>]
```

Parameter Set: Template

```
Get-SCVirtualCOMPort -VMTemplate <Template> [ <CommonParameters>]
```

Parameter Set: VM

```
Get-SCVirtualCOMPort -VM <VM> [ <CommonParameters>]
```

### Detailed Description

The Get-SCVirtualCOMPort cmdlet gets one or both Virtual Machine Manager (VMM) virtual COM port objects from a virtual machine object, a virtual machine template object, or a hardware profile object.

A virtual COM port can connect to a physical port on a virtual machine host server, to a text file, or to a named pipe. Each virtual machine, virtual machine template, and hardware profile contains exactly two COM ports.

### Parameters

#### -All

Retrieves a full list of all subordinate objects independent of the parent object. For example, the command Get-SCVirtualDiskDrive -All retrieves all virtual disk drive objects regardless of the virtual machine object or template object that each virtual disk drive object is associated with.

Aliases	none
Required?	true

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-HardwareProfile<HardwareProfile>**

Specifies a hardware profile object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-VM<VM>**

Specifies a virtual machine object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **-VMTemplate<Template>**

Specifies a VMM template object used to create virtual machines.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **VirtualCOMPort**

## **Notes**

- Requires a VMM virtual machine object, virtual machine template object, or hardware profile object. You can retrieve these objects by using the Get-SCVirtualMachine, Get-SCVMTemplate, and Get-SCHardwareProfile cmdlets, respectively.

## **Examples**

### **1: Get COM ports from a virtual machine.**

The first command gets the virtual machine object named VM01 and stores the object in the \$VM variable.

---

The second command retrieves the virtual COM port objects on VM01 and displays information about these ports to the user.

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM01"  
PS C:\> Get-SCVirtualCOMPort -VM $VM
```

## 2: Get COM ports from a template.

The first command gets the virtual machine template object VMTemplate01 and stores the object in the \$VMTemplate variable.

The second command gets the virtual COM port objects VMTemplate01 and displays information about these ports to the user.

```
PS C:\> $VMTemplate = Get-SCVMTemplate | where { $_.Name -eq "VMTemplate01" }  
PS C:\> Get-SCVirtualCOMPort -VMTemplate $VMTemplate
```

## 3: Get COM ports from a hardware profile.

The first command gets the hardware profile object named NewHWProfile01 and stores the object in the \$HWProfile variable.

The second command gets the virtual COM port objects NewHWProfile01 and displays information about these ports to the user.

```
PS C:\> $HWProfile = Get-SCHardwareProfile | where { $_.Name -eq "NewHWProfile01" }  
PS C:\> Get-SCVirtualCOMPort -HardwareProfile $HWProfile
```

## Related topics

[Get-SCHardwareProfile](#)

[Get-SCVirtualMachine](#)

[Get-SCVMTemplate](#)

[Set-SCVirtualCOMPort](#)



---

# Get-SCVirtualDiskDrive

---

## Get-SCVirtualDiskDrive

Gets a virtual disk drive object on a virtual machine template or on a virtual machine managed by VMM.

### Syntax

Parameter Set: All

```
Get-SCVirtualDiskDrive -All [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: Template

```
Get-SCVirtualDiskDrive -VMTemplate <Template> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: VM

```
Get-SCVirtualDiskDrive -VM <VM> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCVirtualDiskDrive cmdlet gets one or more Virtual Machine Manager (VMM) virtual disk drive objects. These virtual disk drives can be configured on virtual machine templates stored in the library, or on virtual machines either deployed on a host or stored in the library.

### Parameters

#### -All

Retrieves a full list of all subordinate objects independent of the parent object. For example, the command Get-SCVirtualDiskDrive -All retrieves all virtual disk drive objects regardless of the virtual machine object or template object that each virtual disk drive object is associated with.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

---

## **-VM<VM>**

Specifies a virtual machine object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **-VMTemplate<Template>**

Specifies a VMM template object used to create virtual machines.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **VirtualDiskDrive**

## Notes

- Requires a VMM virtual machine template object or a virtual machine object, which can be retrieved by using the Get-SCVMTemplate or Get-SCVirtualMachine cmdlets, respectively.

## Examples

### 1: Get a list of all available virtual disk drives in your VMM environment.

This command gets a list of all virtual disk drives bound to all virtual machines registered to VMM on VMMServer01 . The command displays information about each virtual disk drive to the user.

```
PS C:\> Get-SCVirtualDiskDrive -VMMServer "VMMServer01.Contoso.com" -All
```

### 2: Get virtual disk drives for a specific virtual machine.

The first command gets the virtual machine object named VM02 and stores the object in the \$VM variable.

The second command gets all virtual disk drive objects on VM02 and stores the objects in \$VirtDiskDrive. If, as this example assumes, a virtual machine contains multiple virtual disk drives, each virtual disk drive has connected to it either a virtual hard disk or a pass-through disk.

The last command displays the properties of each virtual disk drive on VM02 to the user, including the name of any virtual hard disks and the path to the physical drive on the host for any pass-through disks.

```
PS C:\> $VM = Get-SCVirtualMachine -VMMServer "VMMServer01.Contoso.com" | where {$_.Name -eq "VM02"}
```

```
PS C:\> $VirtDiskDrive = Get-SCVirtualDiskDrive -VM $VM
```

```
PS C:\> $VirtDiskDrive
```

---

### 3. Count all virtual disk drives, except pass-through disks, on the second slot for both IDE channels.

The first command gets the virtual disk drive objects, excluding pass-through disks, that are connected to the second slot of either IDE channel. Using the '@' symbol and parentheses ensures that the command stores the results in an array in case the command returns a single object or a null value.

The second command displays the number of virtual disk drive objects that match the filter criteria.

```
PS C:\> $VirtDiskDrive = @(Get-SCVirtualDiskDrive -All | where {$_.BusType -eq 'IDE' -and
$_PassThroughDisk -eq $null -and $_.LUN -eq 1 -and ($_.Bus -eq 0 -or $_.Bus -eq 1)})
PS C:\> $VirtDiskDrive.Count
```

### 4: Get virtual disk drives for all virtual machine templates.

The first command gets all virtual machine template objects and stores the objects in \$Templates. Using the "@" symbol and parentheses ensures that the command stores the results in an array in case the command returns a single object or a null value.

The second command passes each virtual machine template object stored in \$Templates to the ForEach cmdlet which gets all disk drive objects for each template. Then the command selects only those virtual disk drive objects with an IDE bus type and passes those objects to the Format-List cmdlet which displays the Name, Bus Type, Bus, and LUN for each virtual disk drive object.

For more information about the standard Windows PowerShell ForEach cmdlet type: "Get-Help ForEach-Object".

```
PS C:\> $Templates = @(Get-SCVMTemplate)
PS C:\> $Templates | ForEach {Get-SCVirtualDiskDrive -Template $_ | where {$_.BusType -eq
"IDE"}} | Format-List Name,BusType,Bus,LUN
```

## Related topics

[Compress-SCVirtualDiskDrive](#)

[Convert-SCVirtualDiskDrive](#)

[Expand-SCVirtualDiskDrive](#)

[New-SCVirtualDiskDrive](#)

[Remove-SCVirtualDiskDrive](#)

[Set-SCVirtualDiskDrive](#)

---

# Get-SCVirtualDVDDrive

---

## Get-SCVirtualDVDDrive

Gets a VMM virtual DVD drive object from a virtual machine, a virtual machine template, or a hardware profile.

### Syntax

Parameter Set: All

```
Get-SCVirtualDVDDrive -All [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: HardwareProfile

```
Get-SCVirtualDVDDrive -HardwareProfile <HardwareProfile> [ <CommonParameters>]
```

Parameter Set: Template

```
Get-SCVirtualDVDDrive -VMTemplate <Template> [ <CommonParameters>]
```

Parameter Set: VM

```
Get-SCVirtualDVDDrive -VM <VM> [ <CommonParameters>]
```

### Detailed Description

The Get-SCVirtualDVDDrive cmdlet gets one or more Virtual Machine Manager (VMM) virtual DVD drive objects from a virtual machine object, a virtual machine template object, or a hardware profile object.

### Parameters

#### -All

Retrieves a full list of all subordinate objects independent of the parent object. For example, the command Get-SCVirtualDiskDrive -All retrieves all virtual disk drive objects regardless of the virtual machine object or template object that each virtual disk drive object is associated with.

Aliases	none
Required?	true
Position?	named
Default Value	none

---

Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-HardwareProfile<HardwareProfile>**

Specifies a hardware profile object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-VM<VM>**

Specifies a virtual machine object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **-VMTemplate<Template>**

Specifies a VMM template object used to create virtual machines.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **VirtualDVDDrive**

## **Notes**

- Requires a VMM virtual machine object, virtual machine template object, or hardware profile object. You can retrieve these objects by using the Get-SCVirtualMachine, Get-SCVMTemplate, or Get-SCHardwareProfile cmdlets, respectively.

## **Examples**

### **1: Get virtual DVD drives from a virtual machine.**

The first command gets the virtual machine object named VM01 and stores the object in the \$VM variable.

The second command gets all DVD drive objects on VM01 and displays information about these virtual DVD drives to the user.

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM01"
```

---

```
PS C:\> Get-SCVirtualDVDDrive -VM $VM
```

## 2: Get virtual DVD drives from a template.

The first command gets all virtual machine template objects stored in the VMM library, selects the template object named VMTemplate01, and then stores the object in the \$Template variable.

The second command gets all virtual DVD drive objects on VMTemplate01 and displays information about these virtual DVD drives to the user.

```
PS C:\> $VMTemplate = Get-SCVMTemplate | where { $_.Name -eq "VMTemplate01" }
```

```
PS C:\> Get-SCVirtualDVDDrive -Template $VMTemplate
```

## 3: Get virtual DVD drives from a hardware profile.

The first command gets all hardware profile objects in the VMM library, selects the profile object named NewHWProfile01, and then stores the object in the \$HWProfile variable.

The second command gets all virtual DVD drive objects on NewHHWProfile01 and displays information about these virtual DVD drives to the user.

```
PS C:\> $HWProfile = Get-SCHardwareProfile | where { $_.Name -eq "NewHWProfile01" }
```

```
PS C:\> Get-SCVirtualDVDDrive -HardwareProfile $HWProfile
```

## Related topics

[Get-SCHardwareProfile](#)

[Get-SCVirtualMachine](#)

[Get-SCVMTemplate](#)

[New-SCVirtualDVDDrive](#)

[Remove-SCVirtualDVDDrive](#)

[Set-SCVirtualDVDDrive](#)



---

# Get-SCVirtualFloppyDisk

---

## Get-SCVirtualFloppyDisk

Gets virtual floppy disk objects from the VMM library.

### Syntax

Parameter Set: All

```
Get-SCVirtualFloppyDisk [-All] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: EquivalentResourceParamSet

```
Get-SCVirtualFloppyDisk -FamilyName <String> [-Release <String> ] [-VMMServer  
<ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ID

```
Get-SCVirtualFloppyDisk [-ID <Guid> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: NameParamSet

```
Get-SCVirtualFloppyDisk -Name <String> [-VMMServer <ServerConnection> ] [  
<CommonParameters>]
```

### Detailed Description

The Get-SCVirtualFloppyDisk cmdlet gets one or more virtual floppy disk objects from the Virtual Machine Manager (VMM) library. The virtual floppy disk file (either a Windows-based .vfd file or a VMware-based .flp file) that a virtual floppy disk object represents is stored on a library server.

### Parameters

#### -All

Retrieves a full list of all subordinate objects independent of the parent object. For example, the command Get-SCVirtualDiskDrive -All retrieves all virtual disk drive objects regardless of the virtual machine object or template object that each virtual disk drive object is associated with.

Aliases	none
Required?	false
Position?	named

---

Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-FamilyName<String>**

Specifies a family name for a physical resource in the VMM library. This value is used in conjunction with Release, Namespace, and Type to establish equivalency among library resources.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-ID<Guid>**

Specifies the numerical identifier (as a globally unique identifier, or GUID) for a specific object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	true

Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Release<String>**

Specifies a string that describes the release of a library resource. VMM automatically creates a release value for every resource imported into the library. After the resource has been imported, the string can be customized.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

---

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **VirtualFloppyDisk**

## Examples

### 1: Get all virtual floppy disks on all VMM library servers.

This command gets all virtual floppy disk objects VMM library on VMMServer01 and then displays information about these virtual floppy disk objects to the user. The virtual floppy disk files themselves are stored in library shares on library servers.

```
PS C:\> Get-SCVirtualFloppyDisk -VMMServer "VMMServer01.Contoso.com"
```

### 2: Get all virtual floppy disks on a specific VMM library server.

This command gets all virtual floppy disk objects stored on LibraryServer01 and displays information about these virtual floppy disk objects to the user.

```
PS C:\> Get-SCVirtualFloppyDisk -VMMServer "VMMServer01.Contoso.com" | where {
$_ .LibraryServer.Name -eq "LibraryServer01.Contoso.com" }
```

### 3: Get all virtual floppy disks with a specific name on any VMM library server.

This command gets all virtual floppy disk objects named BootFloppy.vfd that are stored on any library server managed by VMM, and then displays information about these virtual floppy disk objects to the user.

NOTE: By default, the name of a virtual floppy disk object in the library is the same name (including the extension) as the name of the actual virtual floppy disk file on the library server.

```
PS C:\> Get-SCVirtualFloppyDisk | where { $_.Name -eq "BootFloppy.vfd" }
```

## Related topics

[Get-SCVMMServer](#)

[Remove-SCVirtualFloppyDisk](#)

[Set-SCVirtualFloppyDisk](#)

---

# Get-SCVirtualFloppyDrive

---

## Get-SCVirtualFloppyDrive

Gets a VMM virtual floppy drive objects from a virtual machine, a virtual machine template, or a hardware profile.

### Syntax

Parameter Set: All

```
Get-SCVirtualFloppyDrive -All [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: HardwareProfile

```
Get-SCVirtualFloppyDrive -HardwareProfile <HardwareProfile> [ <CommonParameters>]
```

Parameter Set: Template

```
Get-SCVirtualFloppyDrive -VMTemplate <Template> [ <CommonParameters>]
```

Parameter Set: VM

```
Get-SCVirtualFloppyDrive -VM <VM> [ <CommonParameters>]
```

## Detailed Description

The Get-SCVirtualFloppyDrive cmdlet gets one or more virtual floppy drive objects in a Virtual Machine Manager (VMM) environment from a virtual machine object, a virtual machine template object, or a hardware profile object.

In VMM, each virtual machine, virtual machine template, or hardware profile has one floppy drive. You cannot remove this floppy drive or add any additional floppy drives.

By default, the virtual floppy drive is configured as attached to no media. To configure the virtual floppy drive to use the physical floppy drive on the virtual machine host (typically, drive A:) use the Set-SCVirtualFloppyDrive cmdlet. Alternatively, you can configure the virtual floppy drive to read an existing virtual floppy disk.

## Parameters

### -All

Retrieves a full list of all subordinate objects independent of the parent object. For example, the command Get-SCVirtualDiskDrive -All retrieves all virtual disk drive objects regardless of the virtual machine object or template object that each virtual disk drive object is associated with.

---

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-HardwareProfile<HardwareProfile>**

Specifies a hardware profile object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-VM<VM>**

Specifies a virtual machine object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **-VMTemplate<Template>**

Specifies a VMM template object used to create virtual machines.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **VirtualFloppyDrive**

## **Notes**

- Requires a virtual machine object, virtual machine template object, or hardware profile object, which can be retrieved by using the Get-SCVirtualMachine, Get-SCVMTemplate, or Get-SCHardwareProfile cmdlets, respectively.

---

## Examples

### 1: Get the virtual floppy drive from a virtual machine.

The first command gets the virtual machine object named VM01 and stores the object in the \$VM variable.

The second command gets the virtual floppy drive object on VM01 and displays information about this drive to the user.

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM01"
```

```
PS C:\> Get-SCVirtualFloppyDrive -VM $VM
```

### 2: Get the virtual floppy drive from a virtual machine template.

The first command gets the virtual machine template object named VMTemplate01 and stores the object in the \$VMTemplate variable.

The second command gets the virtual floppy drive object on VMTemplate01 and displays information about the drive to the user.

```
PS C:\> $VMTemplate = Get-SCVMTemplate | where { $_.Name -eq "VMTemplate01" }
```

```
PS C:\> Get-SCVirtualFloppyDrive -VMTemplate $VMTemplate
```

### 3: Get the virtual floppy drive from a hardware profile.

The first command gets the hardware profile named NewHWProfile01 and stores the object in the \$HWProfile variable.

The second command gets the virtual floppy drive object on NewHWProfile01 and displays information about the drive to the user.

```
PS C:\> $HWProfile = Get-SCHardwareProfile | where { $_.Name -eq "NewHWProfile01" }
```

```
PS C:\> Get-SCVirtualFloppyDrive -HardwareProfile $HWProfile
```

## Related topics

[Get-SCHardwareProfile](#)

[Get-SCVirtualMachine](#)

[Get-SCVMTemplate](#)

[Set-SCVirtualFloppyDrive](#)



---

# Get-SCVirtualHardDisk

---

## Get-SCVirtualHardDisk

Gets virtual hard disk objects from a virtual machine, from a template, or as a standalone file stored in the VMM library.

### Syntax

Parameter Set: All

```
Get-SCVirtualHardDisk [-All] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: EquivalentResourceParamSet

```
Get-SCVirtualHardDisk -FamilyName <String> [-Release <String> ] [-VMMServer  
<ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ID

```
Get-SCVirtualHardDisk [-ID <Guid> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: NameParamSet

```
Get-SCVirtualHardDisk -Name <String> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: Template

```
Get-SCVirtualHardDisk -VMTemplate <Template> [-VMMServer <ServerConnection> ] [  
<CommonParameters>]
```

Parameter Set: VM

```
Get-SCVirtualHardDisk -VM <VM> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCVirtualHardDisk gets virtual hard disk objects from a virtual machine, from a template, or as a standalone file stored in the Virtual Machine Manager (VMM) library.

A virtual hard disk can be a Windows-based .vhd file, a Citrix XenServer-based .vhd file, or a VMware-based .vmdk file. A virtual hard disk might be stored as a standalone object in the VMM library, attached to a virtual disk drive on a template, or attached to a virtual disk drive on a deployed or stored virtual machine.

---

## Parameters

### **-All**

Retrieves a full list of all subordinate objects independent of the parent object. For example, the command `Get-SCVirtualDiskDrive -All` retrieves all virtual disk drive objects regardless of the virtual machine object or template object that each virtual disk drive object is associated with.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-FamilyName<String>**

Specifies a family name for a physical resource in the VMM library. This value is used in conjunction with `Release`, `Namespace`, and `Type` to establish equivalency among library resources.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-ID<Guid>**

Specifies the numerical identifier (as a globally unique identifier, or GUID) for a specific object.

Aliases	none
Required?	false
Position?	named
Default Value	none

---

Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Release<String>**

Specifies a string that describes the release of a library resource. VMM automatically creates a release value for every resource imported into the library. After the resource has been imported, the string can be customized.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VM<VM>**

Specifies a virtual machine object.

Aliases	none
Required?	true

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-VMTemplate<Template>**

Specifies a VMM template object used to create virtual machines.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

### **Outputs**

The output type is the type of the objects that the cmdlet emits.

- 
- **VirtualHardDisk**

## Notes

- Requires a VMM virtual machine object or a VMM template object, which can be retrieved by using the Get-SCVirtualMachine cmdlet, or the Get-SCVMTemplate cmdlet, respectively.

## Examples

### 1: Get a virtual hard disk object from the library.

This command gets the virtual hard disk object named VHD01.vhd stored on LibraryServer01 and then stores the object in the \$VHD variable.

```
PS C:\> $VHD = Get-SCVirtualHardDisk -VMMServer "VMMServer01.Contoso.com" | where { $_.Name -eq "VHD01.vhd" -and $_.LibraryServer.Name -eq "LibraryServer01.Contoso.com" }
```

### 2: Get a virtual hard disk object from a virtual machine.

The first command connects to VMMServer01.

The second command gets the virtual machine object named VM01, selects all virtual hard disks on VM01 whose name includes the string "DataDisk," and then stores the returned virtual hard disk objects in the \$VHD variable.

```
PS C:\> Get-SCVMMServer -ComputerName "VMMServer01.Contoso.com"
```

```
PS C:\> $VHD = Get-SCVirtualMachine -Name "VM01" | Get-SCVirtualHardDisk | where { $_.Name -match "DataDisk" }
```

### 3: Get a virtual hard disk object from a specific template.

This command gets the template object named Template01 from the library and displays all virtual hard disk objects on that template to the user.

```
PS C:\> Get-SCVMTemplate -VMMServer "VMMServer01.Contoso.com" | where { $_.Name -eq "Template01" } | Get-SCVirtualHardDisk
```

## Related topics

[Compress-SCVirtualDiskDrive](#)

[Get-SCVirtualDiskDrive](#)

[Set-SCVirtualDiskDrive](#)

[Get-SCVirtualMachine](#)

[Get-SCVMTemplate](#)

[Remove-SCVirtualHardDisk](#)

---

[Set-SCVirtualHardDisk](#)

---

# Get-SCVirtualHardDiskConfiguration

---

## Get-SCVirtualHardDiskConfiguration

Gets the virtual hard disk configuration contained within a virtual machine configuration.

### Syntax

Parameter Set: Default

```
Get-SCVirtualHardDiskConfiguration -VMConfiguration <VMConfiguration> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCVirtualHardDiskConfiguration cmdlet gets the virtual hard disk configuration information that is contained within a virtual machine configuration.

### Parameters

#### **-VMConfiguration<VMConfiguration>**

Specifies a virtual machine configuration object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

#### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
---------	------

---

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **VirtualHardDiskConfiguration**

## Examples

### 1: Get the virtual hard disk configuration for a virtual machine configuration.

The first command gets the service configuration object named Service01 and stores the object in the \$ServiceConfig variable.

The second command gets the computer tier configuration for the service configuration stored in \$ServiceConfig and stores the object in the \$TierConfig variable.

The third command gets the virtual machine configuration for the computer tier configuration stored in \$TierConfig and stores the object in the \$VMConfig variable.

The fourth command gets the virtual hard disk configuration for the first virtual machine configuration stored in \$VMConfig.

The last command displays the properties of the virtual hard disk configuration stored in \$VMConfig to the user.

```
PS C:\> $ServiceConfig = Get-SCServiceConfiguration -Name "Service01"
PS C:\> $TierConfig = Get-SCComputerTierConfiguration -ServiceConfiguration $ServiceConfig
PS C:\> $VMConfig = Get-SCVMConfiguration -ComputerTierConfiguration $TierConfig
PS C:\> $VHDConfig = Get-SCVirtualHardDiskConfiguration -VMConfiguration $VMConfig[0]
PS C:\> $VHDConfig
```



---

## Related topics

[Get-SCComputerTierConfiguration](#)

[Get-SCServiceConfiguration](#)

[Get-SCVMConfiguration](#)

[Set-SCVirtualHardDiskConfiguration](#)

---

# Get-SCVirtualizationManager

---

## Get-SCVirtualizationManager

Gets a VMware vCenter Server object managed by VMM.

### Syntax

Parameter Set: Default

```
Get-SCVirtualizationManager [[-ComputerName] <String> ] [-VMMServer <ServerConnection> ] [ <CommonParameters> ]
```

### Detailed Description

The Get-SCVirtualizationManager cmdlet gets one or VMware vCenter Server objects managed by Virtual Machine Manager (VMM). A vCenter Server is a virtualization manager that typically manages ESX hosts and virtual machines deployed on those hosts.

If a vCenter Server is connected to VMM, you can use this cmdlet to view the properties of the vCenter Server object or to store it in a variable for use by other cmdlets.

### Parameters

#### **-ComputerName<String>**

Specifies the name of a computer that VMM can uniquely identify on your network. Valid formats are: FQDN, IPv4 or IPv6 address, or NetBIOS name.

NOTE: See the examples for a specific cmdlet to determine how that cmdlet specifies the computer name.

Aliases	none
Required?	false
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

### **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **VirtualizationManager**

### **Examples**

#### **1: Display information about each VMware vCenter Server managed by VMM.**

This command retrieves all virtualization manager objects currently associated with VMM from VMMServer01 and displays information about the returned objects.

```
PS C:\> Get-SCVirtualizationManager -VMMServer "VMMServer01.Contoso.com"
```

#### **2: Get a specific VMware vCenter Server managed by VMM.**

This command gets the vCenter Server object named VirtMgrServer02 and displays information about the returned object to the user.

```
PS C:\> Get-SCVirtualizationManager -VMMServer "VMMServer01.Contoso.com" -ComputerName "vCenterServer02.Contoso.com"
```

---

### 3: Get all VMware vCenter Servers that match specified criteria.

The first command gets all virtualization manager objects whose name includes the string "Server" and stores the returned objects in the \$vCenterServers array.

The second command displays information about each vCenter Server object to the user.

```
PS C:\> $vCenterServers = Get-SCVirtualizationManager -VMMServer "VMMServer01.Contoso.com" |  
where {$_.Name -match "Server"}
```

```
PS C:\> $vCenterServers
```

### Related topics

[Add-SCVirtualizationManager](#)

[Read-SCVirtualizationManager](#)

[Remove-SCVirtualizationManager](#)

[Set-SCVirtualizationManager](#)

---

# Get-SCVirtualMachine

---

## Get-SCVirtualMachine

Gets virtual machine objects.

### Syntax

Parameter Set: All

```
Get-SCVirtualMachine [[-Name] <String> ] [-All] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: Cloud

```
Get-SCVirtualMachine [[-Name] <String> ] -Cloud <Cloud> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ComputerTier

```
Get-SCVirtualMachine [[-Name] <String> ] -ComputerTier <ComputerTier> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ID

```
Get-SCVirtualMachine [[-Name] <String> ] [-ID <Guid> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: Service

```
Get-SCVirtualMachine [[-Name] <String> ] -Service <Service> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: VMHostGroup

```
Get-SCVirtualMachine [[-Name] <String> ] -VMHost <Host> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCVirtualMachine cmdlet gets one or more virtual machine objects from the Virtual Machine Manager (VMM) database. A virtual machine can be deployed on a virtual machine host or can be stored in the VMM library.

---

## Parameters

### **-All**

Retrieves a full list of all subordinate objects independent of the parent object. For example, the command `Get-SCVirtualDiskDrive -All` retrieves all virtual disk drive objects regardless of the virtual machine object or template object that each virtual disk drive object is associated with.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Cloud<Cloud>**

Specifies a private cloud object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-ComputerTier<ComputerTier>**

Specifies a computer tier object.

Aliases	none
Required?	true
Position?	named
Default Value	none

---

Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-ID<Guid>**

Specifies the numerical identifier (as a globally unique identifier, or GUID) for a specific object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	false
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Service<Service>**

Specifies a VMM service object.

Aliases	none
Required?	true
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-VMHost<Host>**

Specifies a virtual machine host object. VMM supports Hyper-V hosts, VMware ESX hosts, and Citrix XenServer hosts.

For more information about each type of host, type: "Get-Help Add-SCVMHost -detailed". See the examples for a specific cmdlet to determine how that cmdlet uses this parameter.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).



---

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **VirtualMachine**

## Examples

### 1: Get all virtual machines and display information about each one.

This command gets all virtual machine objects on VMMServer01 and displays information about these virtual machine objects to the user.

```
PS C:\> Get-SCVirtualMachine -VMMServer "VMMServer01.Contoso.com"
```

### 2: Get all virtual machines and display information about specific properties.

This command gets all virtual machine objects on VMMServer01 and displays the values of the specified properties to the user.

```
PS C:\> Get-SCVirtualMachine -VMMServer "VMMServer01.Contoso.com" | Format-List -property Name, Owner, Description, HostName, OperatingSystem, CPUCount, Memory
```

### 3: Get a virtual machine by name that is stored on a specified library server.

The first command connects to VMMServer01.

The second command gets the virtual machine object named VM02 stored on LibraryServer01 and then displays the virtual machine name, the name of the library server, and the status of the virtual machine to the user.

```
PS C:\> Get-SCVMMServer -ComputerName "VMMServer01.Contoso.com"
PS C:\> Get-SCVirtualMachine | where { $_.Name -eq "VM02" -and $_.LibraryServer -eq "LibraryServer01" } | select Name,LibraryServer,Status
```

### 4: Get all virtual machines on the specified host.

The first command connects to VMMServer01.

The second command gets all virtual machine objects deployed on VMHost01 and displays information about these virtual machines to the user.

```
PS C:\> Get-SCVMMServer -ComputerName "VMMServer01.Contoso.com"
PS C:\> Get-SCVirtualMachine -VMHost "VMHost01.Contoso.com"
```

---

## Related topics

[Move-SCVirtualMachine](#)

[New-SCVirtualMachine](#)

[Read-SCVirtualMachine](#)

[Remove-SCVirtualMachine](#)

[Repair-SCVirtualMachine](#)

[Resume-SCVirtualMachine](#)

[Save-SCVirtualMachine](#)

[Set-SCVirtualMachine](#)

[Start-SCVirtualMachine](#)

[Stop-SCVirtualMachine](#)

[Suspend-SCVirtualMachine](#)

---

# Get-SCVirtualNetwork

---

## Get-SCVirtualNetwork

Gets virtual network objects configured on a VMM host.

### Syntax

Parameter Set: NoFilter

```
Get-SCVirtualNetwork [[-Name] <String> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: Cluster

```
Get-SCVirtualNetwork [[-Name] <String> ] -VMHostCluster <HostCluster> [ <CommonParameters>]
```

Parameter Set: FilterByVMHost

```
Get-SCVirtualNetwork [[-Name] <String> ] -VMHost <Host> [ <CommonParameters>]
```

### Detailed Description

The Get-SCVirtualNetwork cmdlet gets one or more virtual network objects configured on a host managed by Virtual Machine Manager (VMM).

For information about virtual networks in VMM, type: "Get-Help New-SCVirtualNetwork -detailed".

### Parameters

#### **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	false
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-VMHost<Host>**

Specifies a virtual machine host object. VMM supports Hyper-V hosts, VMware ESX hosts, and Citrix XenServer hosts.

For more information about each type of host, type: "Get-Help Add-SCVMHost -detailed". See the examples for a specific cmdlet to determine how that cmdlet uses this parameter.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **-VMHostCluster<HostCluster>**

Specifies a VMM host cluster object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)

---

Accept Wildcard Characters?
-----------------------------

false
-------

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **VirtualNetwork**

## Examples

### 1: Get all virtual networks in the VMM database.

The first command gets all virtual network objects on all hosts managed by VMM and stores the virtual network objects in \$VirtualNetworks.

The second command displays a subset of information about each virtual network object in \$VirtualNetworks: the name of each virtual network, the physical host on which each virtual network is configured, and the physical network adapters configured on the host for each virtual network.

```
PS C:\> $VirtualNetworks = Get-SCVirtualNetwork
```

```
PS C:\> $VirtualNetworks | Format-List Name,VMHost,VMHostNetworkadapters
```

### 2: Get all virtual networks on a specific host.

The first command gets the host object named VMHost01 and stores the object in the \$VMHost variable.

The second command gets all virtual network objects on VMHost01 and displays information about each virtual network.

```
PS C:\> $VMHost = Get-SCVMHost -ComputerName "VMHost01.Contoso.com"
```

```
PS C:\> Get-SCVirtualNetwork -VMHost $VMHost
```

### 3: Get a virtual network by name from a specific host.

The first command gets the host object named VMHost02 and stores the object in the \$VMHost variable.

The second command gets the virtual network object named InternalVNet01 from VMHost02 and stores the object in the \$VN variable.

The last command displays information about the virtual network stored in \$VN to the user.

```
PS C:\> $VMHost = Get-SCVMHost -ComputerName "VMHost02.Contoso.com"
```

---

```
PS C:\> $VN = Get-SCVirtualNetwork -VMHost $VMHost -Name "InternalVNet01"  
PS C:\> $VN
```

## Related topics

[New-SCVirtualNetwork](#)

[Remove-SCVirtualNetwork](#)

[Set-SCVirtualNetwork](#)

---

# Get-SCVirtualNetworkAdapter

---

## Get-SCVirtualNetworkAdapter

Gets VMM virtual network adapter objects from a virtual machine, virtual machine template, or hardware profile.

### Syntax

Parameter Set: All

```
Get-SCVirtualNetworkAdapter -All [-ParentTypeVMOrHost] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: HardwareProfile

```
Get-SCVirtualNetworkAdapter -HardwareProfile <HardwareProfile> [ <CommonParameters>]
```

Parameter Set: ID

```
Get-SCVirtualNetworkAdapter [-ID] <Guid> [[-Name] <String> ][ <CommonParameters>]
```

Parameter Set: Template

```
Get-SCVirtualNetworkAdapter -VMTemplate <Template> [ <CommonParameters>]
```

Parameter Set: VM

```
Get-SCVirtualNetworkAdapter -VM <VM> [ <CommonParameters>]
```

Parameter Set: VMHost

```
Get-SCVirtualNetworkAdapter -VMHost <Host> [ <CommonParameters>]
```

## Detailed Description

The Get-SCVirtualNetworkAdapter cmdlet gets one or more virtual network adapter objects from a virtual machine object, a virtual machine template object, or a hardware profile object in a Virtual Machine Manager (VMM) environment.

## Parameters

### -All

Retrieves a full list of all subordinate objects independent of the parent object. For example, the command Get-SCVirtualDiskDrive -All retrieves all virtual disk drive objects regardless of the virtual machine object or template object that each virtual disk drive object is associated with.

---

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-HardwareProfile<HardwareProfile>**

Specifies a hardware profile object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-ID<Guid>**

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-Name<String>**

Aliases	none
---------	------



---

Required?	false
Position?	2
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-ParentTypeVMOrHost**

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-VM<VM>**

Specifies a virtual machine object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-VMHost<Host>**

Aliases	none
Required?	true

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-VMTemplate<Template>**

Specifies a VMM template object used to create virtual machines.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

### **Outputs**

The output type is the type of the objects that the cmdlet emits.

- 
- **VirtualNetworkAdapter**

## Notes

- Requires a VMM virtual machine object, virtual machine template object, or hardware profile object, which can be retrieved by using the Get-SCVirtualMachine, Get-SCVMTemplate, or Get-SCHardwareProfile cmdlets, respectively.

## Examples

### 1: Get virtual network adapters from a virtual machine.

The first command gets the virtual machine object named VM01 and stores the object in the \$VM variable.

The second command gets all virtual network adapter objects on VM01 and displays information about the adapters to the user.

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM01"
```

```
PS C:\> Get-SCVirtualNetworkAdapter -VM $VM
```

### 2: Get virtual network adapters from a virtual machine template.

The first command gets all virtual machine template objects from the VMM library, selects the template object named VMTemplate01, and stores the object in the \$VMTemplate variable.

The last command gets all virtual network adapter objects VMTemplate01 and displays information about these adapters to the user.

```
PS C:\> $VMTemplate = Get-SCVMTemplate | where { $_.Name -eq "VMTemplate01" }
```

```
PS C:\> Get-SCVirtualNetworkAdapter -Template $VMTemplate
```

### 3: Get virtual network adapters from a hardware profile.

The first command gets all hardware profile objects in the VMM library, selects the profile object named NewHWProfile01, and then stores the object in the \$HWProfile variable.

The second command gets all virtual network adapter objects NewHWProfile01 and displays information about these adapters to the user.

```
PS C:\> $HWProfile = Get-SCHardwareProfile | where { $_.Name -eq "NewHWProfile01" }
```

```
PS C:\> Get-SCVirtualNetworkAdapter -HardwareProfile $HWProfile
```

## Related topics

[Get-SCHardwareProfile](#)

[Get-SCVirtualMachine](#)

---

[Get-SCVMTemplate](#)

[Remove-SCVirtualNetworkAdapter](#)

[Set-SCVirtualNetworkAdapter](#)

---

# Get-SCVirtualNetworkAdapterConfiguration

---

## Get-SCVirtualNetworkAdapterConfiguration

Gets the virtual network adapter configuration information contained in a virtual machine configuration.

### Syntax

Parameter Set: Default

```
Get-SCVirtualNetworkAdapterConfiguration -VMConfiguration <VMConfiguration> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCVirtualNetworkAdapterConfiguration cmdlet gets the virtual network adapter configuration information that is contained within a virtual machine configuration.

### Parameters

#### **-VMConfiguration<VMConfiguration>**

Specifies a virtual machine configuration object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

#### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
---------	------

---

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **VirtualNetworkAdapterConfiguration**

## Examples

### 1: Get a virtual network adapter configuration from a virtual machine configuration.

The first command gets the service configuration object named Service01 and stores the object in the \$ServiceConfig variable.

The second command gets the computer tier configuration for the service configuration stored in \$ServiceConfig and stores the object in the \$TierConfig variable.

The third command gets the virtual machine configuration for the computer tier configuration stored in \$TierConfig and stores the object in the \$VMConfig variable.

The fourth command gets the virtual network adapter configuration for the first virtual machine configuration stored in \$VMConfig.

The last command displays the properties of the virtual network adapter configuration stored in \$VMConfig to the user.

```
PS C:\> $ServiceConfig = Get-SCServiceConfiguration -Name "Service01"
PS C:\> $TierConfig = Get-SCComputerTierConfiguration -ServiceConfiguration $ServiceConfig
PS C:\> $VMConfig = Get-SCVMConfiguration -ComputerTierConfiguration $TierConfig
PS C:\> $VNAConfig = Get-SCVirtualNetworkAdapterConfiguration -VMConfiguration $VMConfig[0]
PS C:\> $VNAConfig
```

---

## Related topics

[Get-SCComputerTierConfiguration](#)

[Get-SCServiceConfiguration](#)

[Get-SCVMConfiguration](#)

[Set-SCVirtualNetworkAdapterConfiguration](#)

---

# Get-SCVirtualNetworkAdapterExtensionPortProfile

---

## Get-SCVirtualNetworkAdapterExtensionPortProfile

Gets a virtual network adapter extension port profile object.

### Syntax

Parameter Set: All

```
Get-SCVirtualNetworkAdapterExtensionPortProfile [[-Name] <String> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ByID

```
Get-SCVirtualNetworkAdapterExtensionPortProfile [[-Name] <String> ] -ID <Guid> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ByVirtualSwitchExtension

```
Get-SCVirtualNetworkAdapterExtensionPortProfile [[-Name] <String> ] -VirtualSwitchExtension <VirtualSwitchExtension> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ByVirtualSwitchExtensionManager

```
Get-SCVirtualNetworkAdapterExtensionPortProfile [[-Name] <String> ] -VirtualSwitchExtensionManager <VirtualSwitchExtensionManager> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

## Detailed Description

The Get-SCVirtualNetworkAdapterExtensionPortProfile cmdlet gets one or more virtual network adapter extension port profile objects.

## Parameters

### -ID<Guid>

Specifies the numerical identifier (as a globally unique identifier, or GUID) for a specific object.

Aliases	none
Required?	true



Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	false
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VirtualSwitchExtension<VirtualSwitchExtension>**

Specifies a virtual switch extension object.

To get a virtual switch extension object, use the Get-SCVirtualSwitchExtension cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VirtualSwitchExtensionManager<VirtualSwitchExtensionManager>**

Specifies a virtual switch extension manager object.

To obtain a virtual switch extension manager object, use the Get-SCVirtualSwitchExtensionManager cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Related topics**

[Get-SCVirtualSwitchExtension](#)

[Get-SCVirtualSwitchExtensionManager](#)

---

# Get-SCVirtualNetworkAdapterNativePortProfile

---

## Get-SCVirtualNetworkAdapterNativePortProfile

Gets a virtual network adapter native port profile.

### Syntax

Parameter Set: All

```
Get-SCVirtualNetworkAdapterNativePortProfile [[-Name] <String> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ById

```
Get-SCVirtualNetworkAdapterNativePortProfile [[-Name] <String> ] [-ID <Guid> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCVirtualNetworkAdapterNativePortProfile cmdlet gets one or more virtual network adapter native port profiles. You can retrieve all profiles, or a single profile by its name or ID.

### Parameters

#### **-ID<Guid>**

Specifies the numerical identifier (as a globally unique identifier, or GUID) for a specific object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	false
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## **Examples**

### **1: Get a virtual network adapter native port profile by its name.**

This command retrieves the virtual network adapter native port profile object named VirtualNetworkAdapterNativePortProf01.

```
PS C:\> Get-SCVirtualNetworkAdapterNativePortProfile -Name  
"VirtualNetworkAdapterNativePortProf01"
```

---

## Related topics

[New-SCVirtualNetworkAdapterNativePortProfile](#)

[Remove-SCVirtualNetworkAdapterNativePortProfile](#)

[Set-SCVirtualNetworkAdapterNativePortProfile](#)

---

# Get-SCVirtualNetworkAdapterPortProfileSet

---

## Get-SCVirtualNetworkAdapterPortProfileSet

Gets a virtual network adapter port profile set.

### Syntax

Parameter Set: All

```
Get-SCVirtualNetworkAdapterPortProfileSet [[-Name] <String> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ById

```
Get-SCVirtualNetworkAdapterPortProfileSet [[-Name] <String> ] -ID <Guid> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: LogicalSwitch

```
Get-SCVirtualNetworkAdapterPortProfileSet [[-Name] <String> ] -LogicalSwitch <LogicalSwitch> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCVirtualNetworkAdapterPortProfileSet cmdlet gets one or more virtual network adapter port profile sets. You can get a virtual network adapter port profile set by its name or id, or all virtual network adapter port profile sets for a logical switch.

### Parameters

#### **-ID<Guid>**

Specifies the numerical identifier (as a globally unique identifier, or GUID) for a specific object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-LogicalSwitch<LogicalSwitch>**

Specifies a logical switch object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	false
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## Examples

### 1: Get a virtual network adapter port profile set by its name.

This command gets the virtual network adapter port profile set object named VirtualNetworkAdapterPortProfileSet01 and displays information about the object to the user.

```
PS C:\> Get-SCVirtualNetworkAdapterPortProfileSet -Name "VirtualNetworkAdapterPortProfSet01"
```

### 2: Get all virtual network adapter port profile sets for a logical switch.

The first command gets the logical switch object named LogicalSwitch01 and stores the object in the \$LogSwitich variable.

The second command gets all virtual network adapter port profile sets that use logical switch LogicalSwitch01.

```
PS C:\> $LogSwitch = Get-SCLogicalSwitch -Name "LogicalSwitch01"
```

```
PS C:\> Get-SCVirtualNetworkAdapterPortProfileSet -LogicalSwitch $LogSwitch
```

## Related topics

[New-SCVirtualNetworkAdapterPortProfileSet](#)

[Remove-SCVirtualNetworkAdapterPortProfileSet](#)

[Set-SCVirtualNetworkAdapterPortProfileSet](#)



---

# Get-SCVirtualScsiAdapter

---

## Get-SCVirtualScsiAdapter

Gets a VMM virtual SCSI adapter object from a virtual machine, virtual machine template, or hardware profile.

### Syntax

Parameter Set: All

```
Get-SCVirtualScsiAdapter -All [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: HardwareProfile

```
Get-SCVirtualScsiAdapter -HardwareProfile <HardwareProfile> [ <CommonParameters>]
```

Parameter Set: Template

```
Get-SCVirtualScsiAdapter -VMTemplate <Template> [ <CommonParameters>]
```

Parameter Set: VM

```
Get-SCVirtualScsiAdapter -VM <VM> [ <CommonParameters>]
```

### Detailed Description

The Get-SCVirtualScsiAdapter cmdlet gets one or more virtual SCSI adapter objects used in a Virtual Machine Manager (VMM) environment from a virtual machine object, a virtual machine template object, or from a hardware profile object.

A virtual machine on a Citrix XenServer host always has one virtual SCSI adapter. You cannot remove this adapter or add additional adapters.

### Parameters

#### -All

Retrieves a full list of all subordinate objects independent of the parent object. For example, the command Get-SCVirtualDiskDrive -All retrieves all virtual disk drive objects regardless of the virtual machine object or template object that each virtual disk drive object is associated with.

Aliases	none
Required?	true

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-HardwareProfile<HardwareProfile>**

Specifies a hardware profile object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-VM<VM>**

Specifies a virtual machine object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **-VMTemplate<Template>**

Specifies a VMM template object used to create virtual machines.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **VirtualSCSIAdapter**

## **Notes**

- Requires a VMM virtual machine object, virtual machine template object, or hardware profile object, which can be retrieved by using the Get-SCVirtualMachine, Get-SCVMTemplate, or Get-SCHardwareProfile cmdlets, respectively.

## **Examples**

### **1: Get all virtual SCSI adapters on a virtual machine.**

The first command gets the virtual machine object named VM01 and stores the object in the \$VM variable.

---

The second command gets all virtual SCSI adapter objects on VM01 and displays information about the adapters to the user.

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM01"  
PS C:\> Get-SCVirtualScsiAdapter -VM $VM
```

## 2: Get all virtual SCSI adapters in a virtual machine template.

The first command gets the virtual machine template object named VMTemplate01 from the VMM library and stores the object in the \$VMTemplate variable.

The second command gets all virtual SCSI adapter objects on VMTemplate01 and displays information about the adapters to the user.

```
PS C:\> $VMTemplate = Get-SCVMTemplate | where { $_.Name -eq "VMTemplate01" }  
PS C:\> Get-SCVirtualScsiAdapter -VMTemplate $VMTemplate
```

## 3: Get all virtual SCSI adapters from a hardware profile.

The first command gets the hardware profile object named NewHWProfile01 from the VMM library and stores the object in the \$HWProfile variable.

The second command gets all SCSI adapter objects on NewHWProfile01 and displays information about the adapters to the user.

```
PS C:\> $HWProfile = Get-SCHardwareProfile | where { $_.Name -eq "NewHWProfile01" }  
PS C:\> Get-SCVirtualScsiAdapter -HardwareProfile $HWProfile
```

## Related topics

[Get-SCHardwareProfile](#)

[Get-SCVirtualMachine](#)

[Get-SCVMMServer](#)

[Get-SCVMTemplate](#)

[Remove-SCVirtualScsiAdapter](#)

[Set-SCVirtualScsiAdapter](#)

---

# Get-SCVirtualSwitchExtension

---

## Get-SCVirtualSwitchExtension

Gets a virtual switch extension.

### Syntax

Parameter Set: All

```
Get-SCVirtualSwitchExtension [[-Name] <String> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ByID

```
Get-SCVirtualSwitchExtension [[-Name] <String> ] -ID <Guid> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ByVirtualSwitchExtensionManager

```
Get-SCVirtualSwitchExtension [[-Name] <String> ] -VirtualSwitchExtensionManager <VirtualSwitchExtensionManager> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

## Detailed Description

The Get-SCVirtualSwitchExtension cmdlet gets one or more virtual switch extension objects.

## Parameters

### -ID<Guid>

Specifies the numerical identifier (as a globally unique identifier, or GUID) for a specific object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	false
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VirtualSwitchExtensionManager<VirtualSwitchExtensionManager>**

Specifies a virtual switch extension manager object.

To obtain a virtual switch extension manager object, use the Get-SCVirtualSwitchExtensionManager cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## Examples

### 1: Get a virtual switch extension by its name.

This command gets the virtual switch extension object named VirtualSwitchExtension01 and returns information about the object for the user.

```
PS C:\> Get-SCVirtualSwitchExtension -Name "VirtualSwitchExtension01"
```

## Related topics

[Get-SCVirtualSwitchExtensionManager](#)

---

# Get-SCVirtualSwitchExtensionManager

---

## Get-SCVirtualSwitchExtensionManager

Gets a virtual switch extension manager.

### Syntax

Parameter Set: GlobalList

```
Get-SCVirtualSwitchExtensionManager [[-VirtualSwitchExtensionManagerConnectionString]
<String> ] [-All] [-Manufacturer <String> ] [-Model <String> ] [-VMMServer
<ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ByHostGroup

```
Get-SCVirtualSwitchExtensionManager [[-VirtualSwitchExtensionManagerConnectionString]
<String> ] -VMHostGroup <HostGroup> [-Manufacturer <String> ] [-Model <String> ] [-VMMServer
<ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ByID

```
Get-SCVirtualSwitchExtensionManager [[-VirtualSwitchExtensionManagerConnectionString]
<String> ] -ID <Guid> [-Manufacturer <String> ] [-Model <String> ] [-VMMServer
<ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCVirtualSwitchExtensionManager cmdlet gets one or more virtual extension managers.

### Parameters

#### -All

Retrieves a full list of all subordinate objects independent of the parent object. For example, the command Get-SCVirtualDiskDrive -All retrieves all virtual disk drive objects regardless of the virtual machine object or template object that each virtual disk drive object is associated with.

Aliases	none
Required?	false
Position?	named
Default Value	none



---

Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-ID<Guid>**

Specifies the numerical identifier (as a globally unique identifier, or GUID) for a specific object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Manufacturer<String>**

Specifies the name of the company that manufactured a physical device.

Valid characters include: letters (a-z), numbers (0-9), underscore (\_), hyphen(-), dot(.), and single quote(').

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Model<String>**

Specifies the model of a physical device.

Aliases	none
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VirtualSwitchExtensionManagerConnectionString<String>**

Aliases	none
Required?	false
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMHostGroup<HostGroup>**

Specifies a virtual machine host group object or an array of host group objects.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false

---

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## Related topics

[Add-SCVirtualSwitchExtensionManager](#)

[Read-SCVirtualSwitchExtensionManager](#)

[Remove-SCVirtualSwitchExtensionManager](#)

[Set-SCVirtualSwitchExtensionManager](#)

[Test-SCVirtualSwitchExtensionManager](#)

---

# Get-SCVMCheckpoint

---

## Get-SCVMCheckpoint

Gets virtual machine checkpoint objects from the VMM database.

### Syntax

Parameter Set: Connection

```
Get-SCVMCheckpoint [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ID

```
Get-SCVMCheckpoint [-ID <Guid> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: VM

```
Get-SCVMCheckpoint [-MostRecent] [-VM <VM> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCVMCheckpoint cmdlet gets one or more virtual machine checkpoint objects from the Virtual Machine Manager (VMM) database.

A virtual machine checkpoint is a point-in-time "snapshot" of a virtual machine. You can use the checkpoint to revert a virtual machine to a previous state. For more information about VMM checkpoints, type: "Get-Help New-VMCheckpoint -detailed".

### Parameters

#### **-ID<Guid>**

Specifies the numerical identifier (as a globally unique identifier, or GUID) for a specific object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false

---

Accept Wildcard Characters?	false
-----------------------------	-------

### **-MostRecent**

Specifies the most recent VMM virtual machine checkpoint object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VM<VM>**

Specifies a virtual machine object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)

---

Accept Wildcard Characters?	false
-----------------------------	-------

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- VMCheckpoint

## Examples

### 1: Get all existing checkpoints for each virtual machine.

This command gets all existing checkpoint objects for each virtual machine managed by VMMServer01 and displays information about these checkpoint objects to the user.

```
PS C:\> Get-SCVMCheckpoint -VMMServer "VMMServer01.Contoso.com"
```

### 2: Get all checkpoints for one or more virtual machines with a specific name.

The first command gets all checkpoint objects for virtual machine VM01 and stores the objects in the \$Checkpoints object array.

The second command displays information about the checkpoint objects in \$Checkpoints.

```
PS C:\> $Checkpoints = Get-SCVMCheckpoint -VM "VM01"
```

```
PS C:\> $Checkpoints
```

### 3: Get the hardware profile of the most recently created checkpoint on a VM deployed on a Hyper-V host.

The first command gets the virtual machine object named VM01 and stores the object in the \$VM variable.

The second command gets the most recent checkpoint object created for VM01 and stores the object in the \$Checkpoint variable.

The last command displays information about the hardware profile for checkpoint stored in \$Checkpoint (the most recent checkpoint object created for VM01).

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM01"
```

```
PS C:\> $Checkpoint = $VM | Get-SCVMCheckpoint -MostRecent
```

---

```
PS C:\> $Checkpoint.CheckpointHWProfile
```

## 4: Display the .NET type, events, methods, and properties for checkpoint objects.

The first command gets all checkpoint objects on VMMServer01 and stores the objects in the \$Checkpoints object array.

The second command passes each checkpoint object in \$Checkpoints to the Get-Member cmdlet, which displays the .NET TypeName and the Name, MemberType, and Definition for each event, method, and property associated with this object type.

The last command is the same as the second command except that it pipes the output to the Format-List cmdlet so that you can see the complete definition for each event, method, and property for the checkpoint object type.

```
PS C:\> $Checkpoints = Get-SCVMCheckpoint -VMMServer "VMMServer01.Contoso.com"
```

```
PS C:\> $Checkpoints | Get-Member
```

```
PS C:\> $Checkpoints | Get-Member | Format-List
```

## Related topics

[New-SCVMCheckpoint](#)

[Remove-SCVMCheckpoint](#)

[Restore-SCVMCheckpoint](#)

[Set-SCVMCheckpoint](#)

---

# Get-SCVMConfiguration

---

## Get-SCVMConfiguration

Gets the virtual machine configuration information for a service configuration or computer tier configuration.

### Syntax

Parameter Set: All

```
Get-SCVMConfiguration [-All] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ID

```
Get-SCVMConfiguration [-ID <Guid> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ServiceConfig

```
Get-SCVMConfiguration -ServiceConfiguration <ServiceConfiguration> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: TierConfig

```
Get-SCVMConfiguration -ComputerTierConfiguration <BaseComputerTierConfiguration> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

## Detailed Description

The Get-SCVMConfiguration cmdlet gets virtual machine configuration information for a service configuration or computer tier configuration.

## Parameters

### -All

Retrieves a full list of all subordinate objects independent of the parent object. For example, the command `Get-SCVirtualDiskDrive -All` retrieves all virtual disk drive objects regardless of the virtual machine object or template object that each virtual disk drive object is associated with.

Aliases	none
Required?	false
Position?	named



---

Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-ComputerTierConfiguration<BaseComputerTierConfiguration>**

Specifies a computer tier configuration object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-ID<Guid>**

Specifies the numerical identifier (as a globally unique identifier, or GUID) for a specific object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-ServiceConfiguration<ServiceConfiguration>**

Specifies a service configuration object.

Aliases	none
Required?	true
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **VMConfiguration**

## **Examples**

### **1: Get all configuration information for a virtual machine within a computer tier configuration.**

The first command gets the service configuration object named Service01 and stores the object in the \$ServiceConfig variable.

The second command gets the computer tier configuration for the service configuration stored in \$ServiceConfig and stores the object in the \$TierConfig variable.

The third command gets the virtual machine configuration for the computer tier configuration stored in \$TierConfig and stores the object in the \$VMConfig variable.

The last command displays the properties of the virtual machine configuration stored in \$VMConfig to the user.

---

```
PS C:\> $ServiceConfig = Get-SCServiceConfiguration -Name "Service01"
PS C:\> $TierConfig = Get-SCComputerTierConfiguration -ServiceConfiguration $ServiceConfig
PS C:\> $VMConfig = Get-SCVMConfiguration -ComputerTierConfiguration $TierConfig
PS C:\> $VMConfig
```

## 2: Get all configuration information for a virtual machine within a service configuration

The first command gets the service configuration named Service01 and stores the object in the \$ServiceConfig variable.

The second command gets all virtual machine configurations for the service configuration stored in \$ServiceConfig and stores the objects in the \$VMConfigs variable.

The last command displays the properties of the virtual machine configurations stored in \$VMConfigs to the user.

```
PS C:\> $ServiceConfig = Get-SCServiceConfiguration -Name "Service01"
PS C:\> $VMConfigs = Get-SCVMConfiguration -ServiceConfiguration $ServiceConfig
PS C:\> $VMConfigs
```

### Related topics

[Get-SCComputerTierConfiguration](#)

[Get-SCServiceConfiguration](#)

[Set-SCVMConfiguration](#)

---

# Get-SCVMConsoleInformation

---

## Get-SCVMConsoleInformation

Displays VMM console information about a virtual machine.

### Syntax

Parameter Set: Default

```
Get-SCVMConsoleInformation [-VM] <VM> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCVMConsoleInformation cmdlet displays Virtual Machine Manager (VMM) console information about a virtual machine.

### Parameters

#### **-VM<VM>**

Specifies a virtual machine object.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

#### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
---------	------

---

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Examples

### 1: Get VMM console information for a specified virtual machine.

The first command gets the virtual machine object named VM01 and stores the object in the \$VM variable.

The second command displays the VMM console information about VM01 to the user.

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM01"
```

```
PS C:\> Get-SCVMConsoleInformation -VM $VM
```

## Related topics

[Get-SCVirtualMachine](#)

---

# Get-SCVMHost

---

## Get-SCVMHost

Gets virtual machine host objects from the Virtual Machine Manager database.

### Syntax

Parameter Set: Connection

```
Get-SCVMHost [[-ComputerName] <String> ] [-ID <Guid> ] [-VMMServer <ServerConnection> ] [ <CommonParameters> ]
```

Parameter Set: VMHostCluster

```
Get-SCVMHost [[-ComputerName] <String> ] -VMHostCluster <HostCluster> [ <CommonParameters> ]
```

Parameter Set: VMHostGroup

```
Get-SCVMHost [[-ComputerName] <String> ] -VMHostGroup <HostGroup> [ <CommonParameters> ]
```

### Detailed Description

Gets one or more virtual machine host objects from the Virtual Machine Manager (VMM) database. Virtual machine hosts are physical computers that are managed by VMM on which you can deploy virtual machines.

Virtual Machine Manager supports the following types of hosts:

- Hyper-V hosts
- VMware ESX hosts
- Citrix XenServer hosts

For more information about virtual machine hosts in Virtual Machine Manager, type: "Get-Help Add-VMHost -detailed".

### Parameters

#### **-ComputerName<String>**

Specifies the name of a computer that VMM can uniquely identify on your network. Valid formats are: FQDN, IPv4 or IPv6 address, or NetBIOS name.

NOTE: See the examples for a specific cmdlet to determine how that cmdlet specifies the computer name.

Aliases	none
Required?	false
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-ID<Guid>**

Specifies the numerical identifier (as a globally unique identifier, or GUID) for a specific object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMHostCluster<HostCluster>**

Specifies a VMM host cluster object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-VMHostGroup<HostGroup>**

Specifies a virtual machine host group object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **Host**

## **Examples**

### **1: Get all hosts managed by the specified VMM server.**

This command gets all host objects for all managed by VMMServer01 and displays the host properties to the user.

```
PS C:\> Get-SCVMHost -VMMServer "VMMServer01.Contoso.com"
```



---

## 2: Get a host by name.

This command gets the host object named VMHost01 in the Contoso.com domain and displays the host properties to the user.

```
PS C:\> Get-SCVMHost -ComputerName "VMHost01.Contoso.com"
```

## 3: Get all hosts in a specific host group and display information about them to the user.

The first command gets the host group object named HostGroup01 from VMMServer01 and stores the object in the \$HostGroup variable.

The second command gets all host objects in the host group stored in \$HostGroup and stores the objects in the \$HostsInHG variable.

The last command uses the pipeline operator to pass all host objects stored in \$HostsInHG to the Format-Table cmdlet, which displays the name of each host and the virtual machines deployed on that host in a table.

```
PS C:\> $HostGroup = Get-SCVMHostGroup -Name "HostGroup01" -VMMServer  
"VMMServer01.Contoso.com"
```

```
PS C:\> $HostsInHG = Get-SCVMHost -VMHostGroup $HostGroup
```

```
PS C:\> $HostsInHG | Format-Table -property Name, VMs
```

## 4: Get all hosts in a specific host cluster and display information about them to the user.

The first command gets the host cluster object named Cluster01 from VMMServer01 and stores the object in the \$Cluster variable.

The second command gets all host objects in Cluster01 and stores the objects in the \$HostsInCluster variable.

The last command passes all host objects stored in \$HostsInCluster to the Format-Table cmdlet, which displays the name and virtualization platform of each host in Cluster01.

```
PS C:\> $Cluster = Get-SCVMHostCluster -Name "Cluster01.Contoso.com" -VMMServer  
"VMMServer01.Contoso.com"
```

```
PS C:\> $HostsInCluster = Get-SCVMHost -VMHostCluster $Cluster
```

```
PS C:\> $HostsInCluster | Format-Table -property Name, VirtualizationPlatform
```

## 5: Get a specific host located on a perimeter network by its IP address.

The first command gets the host object located on a perimeter network whose IP address is 10.199.53.5 from VMMServer01 and stores the object in the \$VMHost variable.

---

The second command uses the pipeline operator to pass the host object in \$VMHost to the Select-Object cmdlet, which displays the computer name and operating system for the host.

```
PS C:\> $VMHost = Get-VMHost -ComputerName 10.199.53.5 -VMMServer "VMMServer01.Contoso.com"
```

```
PS C:\> $VMHost | Select-Object -property ComputerName, OperatingSystem
```

## Related topics

[Add-SCVMHost](#)

[Move-SCVMHost](#)

[Read-SCVMHost](#)

[Remove-SCVMHost](#)

[Set-SCVMHost](#)

---

# Get-SCVMHostCluster

---

## Get-SCVMHostCluster

Gets a host cluster object.

### Syntax

Parameter Set: Connection

```
Get-SCVMHostCluster [[-Name] <String> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: VMHostGroup

```
Get-SCVMHostCluster [[-Name] <String> ] -VMHostGroup <HostGroup> [ <CommonParameters>]
```

### Detailed Description

The Get-SCVMHostCluster cmdlet gets one or more host cluster objects from the Virtual Machine Manager (VMM) database.

### Parameters

#### **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	false
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

#### **-VMHostGroup<HostGroup>**

Specifies a virtual machine host group object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **VMHostCluster**

## **Examples**

### **1: Get all host clusters managed by a VMM server.**

This command gets all host cluster objects on VMMServer01 and displays information about each host cluster to the user.

```
PS C:\> Get-SCVMHostCluster -VMMServer "VMMServer01.Contoso.com"
```

---

## 2: Get all host clusters managed by a VMM server and display the host cluster name and virtualization platform for each cluster.

This command gets all host cluster objects on VMMServer01 and passes each object to the select-object cmdlet which displays the name and virtualization platform for each host cluster.

```
PS C:\> Get-SCVMHostCluster -VMMServer "VMMServer01.Contoso.com" | select -Property Name, VirtualizationPlatform
```

## 3: Get a host cluster by name.

This command gets the host cluster object named VMHostCluster03 and displays information about the cluster to the user.

```
PS C:\> Get-SCVMHostCluster -Name "VMHostCluster03.Contoso.com"
```

## 4: Display the object type, methods, and properties for a host cluster managed by VMM.

This command gets the host cluster objects on VMMServer01 and then passes a host cluster object to the Get-Member cmdlet, which displays the .NET type for a host cluster object and the list of events, methods, and properties associated with a host cluster object.

```
PS C:\> Get-SCVMHostCluster -VMMServer "VMMServer01.Contoso.com" | Get-Member
```

## Related topics

[Add-SCVMHostCluster](#)

[Find-SCCluster](#)

[Install-SCVMHostCluster](#)

[Move-SCVMHostCluster](#)

[Read-SCVMHostCluster](#)

[Remove-SCVMHostCluster](#)

[Set-SCVMHostCluster](#)

[Test-SCVMHostCluster](#)

[Uninstall-SCVMHostCluster](#)

---

# Get-SCVMHostGroup

---

## Get-SCVMHostGroup

Gets a host group object from the VMM database.

### Syntax

Parameter Set: Connection

```
Get-SCVMHostGroup [[-Name] <String> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ID

```
Get-SCVMHostGroup [[-Name] <String> ] -ID <Guid> [ <CommonParameters>]
```

Parameter Set: ParentHostGroup

```
Get-SCVMHostGroup [[-Name] <String> ] -ParentHostGroup <HostGroup> [ <CommonParameters>]
```

### Detailed Description

The Get-SCVMHostGroup cmdlet gets one or more host group objects from the Virtual Machine Manager (VMM) database.

For more information about host groups, type: "Get-Help New-VMHostGroup -detailed".

### Parameters

#### -ID<Guid>

Specifies the numerical identifier (as a globally unique identifier, or GUID) for a specific object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	false
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-ParentHostGroup<HostGroup>**

Specifies the parent host group that contains one or more hosts, host groups, or host clusters.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Examples

### 1: Get all host groups at the specified path.

This command gets the host groups located at host path "All Hosts\HostGroup01" and displays information about these host groups to the user.

```
PS C:\> Get-SCVMHostGroup | where { $_.Path -eq "All Hosts\HostGroup01" }
```

### 2: Display the name and path properties for all host groups.

This command gets all host group objects from VMMServer01, selects the name and host group path properties, and displays those properties to the user.

```
PS C:\> Get-SCVMHostGroup -VMMServer "VMMServer01.Contoso.com" | select -Property Name,Path
```

## Related topics

[Move-SCVMHostGroup](#)

[New-SCVMHostGroup](#)

[Remove-SCVMHostGroup](#)

[Set-SCVMHostGroup](#)



---

# Get-SCVMHostNetworkAdapter

---

## Get-SCVMHostNetworkAdapter

Gets physical network adapter objects on a VMM host.

### Syntax

Parameter Set: NoFilter

```
Get-SCVMHostNetworkAdapter [[-Name] <String> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: FilterByVMHost

```
Get-SCVMHostNetworkAdapter [[-Name] <String> ] -VMHost <Host> [ <CommonParameters>]
```

### Detailed Description

The Get-SCVMHostNetworkAdapter cmdlet gets one or more physical network adapter objects on a host managed by Virtual Machine Manager (VMM).

### Parameters

#### **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	false
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-VMHost<Host>**

Specifies a virtual machine host object. VMM supports Hyper-V hosts, VMware ESX hosts, and Citrix XenServer hosts.

For more information about each type of host, type: "Get-Help Add-SCVMHost -detailed". See the examples for a specific cmdlet to determine how that cmdlet uses this parameter.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **HostNetworkAdapter**

---

## Examples

### 1: Get all physical network adapters on the specified host.

The first command gets the host object named VMHost01 and stores the object in the \$VMHost variable.

The second command gets all physical network adapter objects from VMHost01 and then stores the objects in the \$HostAdapter variable.

The third command displays the name and connection state for each adapter.

```
PS C:\> $VMHost = Get-SCVMHost -ComputerName "VMHost01.Contoso.com"
PS C:\> $HostAdapter = Get-SCVMHostNetworkAdapter -VMHost $VMHost
PS C:\> $HostAdapter | select -property Name, ConnectionState
```

### 2: Get all physical network adapters in the VMM database.

This command gets all physical network adapter objects on all hosts managed by the VMM server and displays each adapter's name, its MAC address, its host name, and its maximum bandwidth.

```
PS C:\> Get-SCVMHostNetworkAdapter | Format-List Name, MacAddress, VMHost, MaxBandwidth
```

### 3: Get a physical network adapter by name from a specific host.

The first command gets the host object named VMHost01 and stores the object in the \$VMHost variable.

The second command gets the network adapter object named HostAdapter01 from VMHost01 and stores the object in the \$HostAdapter variable.

The third command passes the adapter object stored in \$HostAdapter to the Format-List cmdlet, which displays the name, whether or not the virtual LAN is enabled, and the current value for the VLAN mode (either Trunk or Access).

```
PS C:\> $VMHost = Get-SCVMHost -ComputerName "VMHost01.Contoso.com"
PS C:\> $HostAdapter = Get-SCVMHostNetworkAdapter -VMHost $VMHost -Name "HostAdapter01"
PS C:\> $HostAdapter | Format-List -property Name,VLANEnabled,VLANMode
```

### 4: Get each host network adapter that includes "Broadcom" in its name.

Gets host network adapter objects from VMMServer01 that include the string "Broadcom" in their name. and then displays the name and IP addresses for each adapter.

```
PS C:\> Get-SCVMHostNetworkAdapter -VMMServer "VMMServer01.Contoso.com" | where { $_.Name - match "Broadcom" } | Format-List -property Name,IPAddresses
```

---

## Related topics

[Add-SCVMHostNetworkAdapter](#)

[Remove-SCVMHostNetworkAdapter](#)

[Set-SCVMHostNetworkAdapter](#)

---

# Get-SCVMHostNetworkAdapterProfile

---

## Get-SCVMHostNetworkAdapterProfile

Gets a host network adapter profile.

### Syntax

Parameter Set: GetById

```
Get-SCVMHostNetworkAdapterProfile -ID <Guid> [-VMMServer <ServerConnection> ] [ <CommonParameters> ]
```

### Detailed Description

The Get-SCVMHostNetworkAdapterProfile cmdlet gets a host network adapter profile by its ID.

### Parameters

#### **-ID<Guid>**

Specifies the numerical identifier (as a globally unique identifier, or GUID) for a specific object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

#### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
---------	------

---

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **HostNetworkAdapterProfile[]**

## Examples

### 1: Get a host network adapter profile by its ID.

This command gets the network adapter profile object with the ID of 259f47c7-c5a9-429d-a421-d232f9b34991.

```
PS C:\> Get-SCVMHostNetworkAdapterProfile -ID "259f47c7-c5a9-429d-a421-d232f9b34991"
```

## Related topics

[New-SCVMHostNetworkAdapterProfile](#)

[Remove-SCVMHostNetworkAdapterProfile](#)

---

# Get-SCVMHostProfile

---

## Get-SCVMHostProfile

Gets a host profile.

### Syntax

Parameter Set: All

```
Get-SCVMHostProfile [-All] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ID

```
Get-SCVMHostProfile [-ID <Guid> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: NameParameter

```
Get-SCVMHostProfile [[-Name] <String> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

## Detailed Description

The Get-SCVMHostProfile cmdlet gets one or more host profiles.

## Parameters

### -All

Retrieves a full list of all subordinate objects independent of the parent object. For example, the command Get-SCVirtualDiskDrive -All retrieves all virtual disk drive objects regardless of the virtual machine object or template object that each virtual disk drive object is associated with.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-ID<Guid>**

Specifies the numerical identifier (as a globally unique identifier, or GUID) for a specific object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	false
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false



---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **HostProfile[]**
- **HostProfile**

## Examples

### 1: Get all host profiles.

This command returns information about all host profiles in the VMM library.

```
PS C:\> Get-SCVMHostProfile
```

### 2: Get a host profile by its name.

This command returns information about the host profile named HostProfile01.

```
PS C:\> Get-SCVMHostProfile -Name "HostProfile01"
```

## Related topics

[New-SCVMHostProfile](#)

[Remove-SCVMHostProfile](#)

[Set-SCVMHostProfile](#)

---

# Get-SCVMHostRating

---

## Get-SCVMHostRating

Calculates the placement rating for one or more hosts managed by VMM on which you might want to deploy a specific virtual machine.

### Syntax

Parameter Set: FromVMClouds

```
Get-SCVMHostRating -Cloud <Cloud[]> -VM <VM> [-CPUExpectedUtilizationPercent <UInt16> ] [-CPUPriority <UInt16> ] [-DiscoveryID <Guid> ] [-DiskIOExpectedCountPerSecond <Int32> ] [-DiskPriority <UInt16> ] [-HighlyAvailable <Boolean> ] [-IsCloudOnlyRating] [-IsMigration] [-JobVariable <String> ] [-MemoryPriority <UInt16> ] [-NetworkPriority <UInt16> ] [-NetworkUtilizationExpectedMbps <Int32> ] [-PlacementGoal {LoadBalance | Consolidate} ] [-ReturnFirstSuitableHost] [-RunAsynchronously] [-UseDefaultPath] [-VMMServer <ServerConnection> ] [-VMName <String> ] [ <CommonParameters>]
```

Parameter Set: FromVMHostGroups

```
Get-SCVMHostRating -VM <VM> -VMHostGroup <HostGroup[]> [-CPUExpectedUtilizationPercent <UInt16> ] [-CPUPriority <UInt16> ] [-DiscoveryID <Guid> ] [-DiskIOExpectedCountPerSecond <Int32> ] [-DiskPriority <UInt16> ] [-HighlyAvailable <Boolean> ] [-IsMigration] [-JobVariable <String> ] [-MemoryPriority <UInt16> ] [-NetworkPriority <UInt16> ] [-NetworkUtilizationExpectedMbps <Int32> ] [-PlacementGoal {LoadBalance | Consolidate} ] [-ReturnFirstSuitableHost] [-RunAsynchronously] [-UseDefaultPath] [-VMMServer <ServerConnection> ] [-VMName <String> ] [ <CommonParameters>]
```

Parameter Set: FromVMHosts

```
Get-SCVMHostRating -VM <VM> -VMHost <Host[]> [-CPUExpectedUtilizationPercent <UInt16> ] [-CPUPriority <UInt16> ] [-DiscoveryID <Guid> ] [-DiskIOExpectedCountPerSecond <Int32> ] [-DiskPriority <UInt16> ] [-HighlyAvailable <Boolean> ] [-IsMigration] [-JobVariable <String> ] [-MemoryPriority <UInt16> ] [-NetworkPriority <UInt16> ] [-NetworkUtilizationExpectedMbps <Int32> ] [-PlacementGoal {LoadBalance | Consolidate} ] [-ReturnFirstSuitableHost] [-RunAsynchronously] [-UseDefaultPath] [-VMMServer <ServerConnection> ] [-VMName <String> ] [ <CommonParameters>]
```

Parameter Set: NewVMClouds

```
Get-SCVMHostRating -Cloud <Cloud[]> -DiskSpaceGB <UInt16> -HardwareProfile <HardwareProfile> -VMName <String> [-CPUExpectedUtilizationPercent <UInt16> ] [-CPUPriority <UInt16> ] [-DiscoveryID <Guid> ] [-DiskIOExpectedCountPerSecond <Int32> ] [-DiskPriority <UInt16> ] [-IsCloudOnlyRating] [-IsMigration] [-JobGroup <Guid> ] [-JobVariable <String> ] [-LUNCountRequirement <UInt32> ] [-MemoryPriority <UInt16> ] [-NetworkPriority <UInt16> ] [-NetworkUtilizationExpectedMbps <Int32> ] [-OperatingSystem <OperatingSystem> ] [-PlacementGoal {LoadBalance | Consolidate} ] [-ReturnFirstSuitableHost] [-RunAsynchronously] [-UseDefaultPath] [-VirtualizationPlatform {Unknown | VirtualServer | HyperV | VMWareVC | VMWareESX | XENServer} ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: NewVMCloudsConfig

---

Get-SCVMHostRating -Cloud <Cloud[]> -VMConfiguration <BaseVMConfiguration> [-CPUExpectedUtilizationPercent <UInt16> ] [-CPUPriority <UInt16> ] [-DiscoveryID <Guid> ] [-DiskIOExpectedCountPerSecond <Int32> ] [-DiskPriority <UInt16> ] [-DiskSpaceGB <UInt16> ] [-IsCloudOnlyRating] [-IsMigration] [-JobVariable <String> ] [-MemoryPriority <UInt16> ] [-NetworkPriority <UInt16> ] [-NetworkUtilizationExpectedMbps <Int32> ] [-PlacementGoal {LoadBalance | Consolidate} ] [-ReturnFirstSuitableHost] [-RunAsynchronously] [-UseDefaultPath] [-VirtualizationPlatform {Unknown | VirtualServer | HyperV | VMWareVC | VMWareESX | XENServer} ] [-VMMServer <ServerConnection> ] [ <CommonParameters> ]

Parameter Set: NewVMCloudsTemplate

Get-SCVMHostRating -Cloud <Cloud[]> -DiskSpaceGB <UInt16> -VMName <String> -VMTemplate <Template> [-CPUExpectedUtilizationPercent <UInt16> ] [-CPUPriority <UInt16> ] [-DiscoveryID <Guid> ] [-DiskIOExpectedCountPerSecond <Int32> ] [-DiskPriority <UInt16> ] [-IsCloudOnlyRating] [-IsMigration] [-JobVariable <String> ] [-MemoryPriority <UInt16> ] [-NetworkPriority <UInt16> ] [-NetworkUtilizationExpectedMbps <Int32> ] [-PlacementGoal {LoadBalance | Consolidate} ] [-ReturnFirstSuitableHost] [-RunAsynchronously] [-UseDefaultPath] [-VirtualizationPlatform {Unknown | VirtualServer | HyperV | VMWareVC | VMWareESX | XENServer} ] [-VMMServer <ServerConnection> ] [ <CommonParameters> ]

Parameter Set: NewVMHostGroups

Get-SCVMHostRating -DiskSpaceGB <UInt16> -HardwareProfile <HardwareProfile> -VMHostGroup <HostGroup[]> -VMName <String> [-CPUExpectedUtilizationPercent <UInt16> ] [-CPUPriority <UInt16> ] [-DiscoveryID <Guid> ] [-DiskIOExpectedCountPerSecond <Int32> ] [-DiskPriority <UInt16> ] [-IsMigration] [-JobGroup <Guid> ] [-JobVariable <String> ] [-LUNCountRequirement <UInt32> ] [-MemoryPriority <UInt16> ] [-NetworkPriority <UInt16> ] [-NetworkUtilizationExpectedMbps <Int32> ] [-OperatingSystem <OperatingSystem> ] [-PlacementGoal {LoadBalance | Consolidate} ] [-ReturnFirstSuitableHost] [-RunAsynchronously] [-UseDefaultPath] [-VirtualizationPlatform {Unknown | VirtualServer | HyperV | VMWareVC | VMWareESX | XENServer} ] [-VMMServer <ServerConnection> ] [ <CommonParameters> ]

Parameter Set: NewVMHostGroupsConfig

Get-SCVMHostRating -VMConfiguration <BaseVMConfiguration> -VMHostGroup <HostGroup[]> [-CPUExpectedUtilizationPercent <UInt16> ] [-CPUPriority <UInt16> ] [-DiscoveryID <Guid> ] [-DiskIOExpectedCountPerSecond <Int32> ] [-DiskPriority <UInt16> ] [-DiskSpaceGB <UInt16> ] [-IsMigration] [-JobVariable <String> ] [-MemoryPriority <UInt16> ] [-NetworkPriority <UInt16> ] [-NetworkUtilizationExpectedMbps <Int32> ] [-PlacementGoal {LoadBalance | Consolidate} ] [-ReturnFirstSuitableHost] [-RunAsynchronously] [-UseDefaultPath] [-VirtualizationPlatform {Unknown | VirtualServer | HyperV | VMWareVC | VMWareESX | XENServer} ] [-VMMServer <ServerConnection> ] [ <CommonParameters> ]

Parameter Set: NewVMHostGroupsTemplate

Get-SCVMHostRating -DiskSpaceGB <UInt16> -VMHostGroup <HostGroup[]> -VMName <String> -VMTemplate <Template> [-CPUExpectedUtilizationPercent <UInt16> ] [-CPUPriority <UInt16> ] [-DiscoveryID <Guid> ] [-DiskIOExpectedCountPerSecond <Int32> ] [-DiskPriority <UInt16> ] [-IsMigration] [-JobVariable <String> ] [-MemoryPriority <UInt16> ] [-NetworkPriority <UInt16> ] [-NetworkUtilizationExpectedMbps <Int32> ] [-PlacementGoal {LoadBalance | Consolidate} ] [-ReturnFirstSuitableHost] [-RunAsynchronously] [-UseDefaultPath] [-VirtualizationPlatform {Unknown | VirtualServer | HyperV | VMWareVC | VMWareESX | XENServer} ] [-VMMServer <ServerConnection> ] [ <CommonParameters> ]

Parameter Set: NewVMHosts

Get-SCVMHostRating -DiskSpaceGB <UInt16> -HardwareProfile <HardwareProfile> -VMHost <Host[]>

---

```
-VMName <String> [-CPUExpectedUtilizationPercent <UInt16> ] [-CPUPriority <UInt16> ] [-DiscoveryID <Guid> ] [-DiskIOExpectedCountPerSecond <Int32> ] [-DiskPriority <UInt16> ] [-IsMigration] [-JobGroup <Guid> ] [-JobVariable <String> ] [-LUNCountRequirement <UInt32> ] [-MemoryPriority <UInt16> ] [-NetworkPriority <UInt16> ] [-NetworkUtilizationExpectedMbps <Int32> ] [-OperatingSystem <OperatingSystem> ] [-PlacementGoal {LoadBalance | Consolidate} ] [-ReturnFirstSuitableHost] [-RunAsynchronously] [-UseDefaultPath] [-VirtualizationPlatform {Unknown | VirtualServer | HyperV | VMWareVC | VMWareESX | XENServer} ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: NewVMHostsConfig

```
Get-SCVMHostRating -VMConfiguration <BaseVMConfiguration> -VMHost <Host[]> [-CPUExpectedUtilizationPercent <UInt16> ] [-CPUPriority <UInt16> ] [-DiscoveryID <Guid> ] [-DiskIOExpectedCountPerSecond <Int32> ] [-DiskPriority <UInt16> ] [-DiskSpaceGB <UInt16> ] [-IsMigration] [-JobVariable <String> ] [-MemoryPriority <UInt16> ] [-NetworkPriority <UInt16> ] [-NetworkUtilizationExpectedMbps <Int32> ] [-PlacementGoal {LoadBalance | Consolidate} ] [-ReturnFirstSuitableHost] [-RunAsynchronously] [-UseDefaultPath] [-VirtualizationPlatform {Unknown | VirtualServer | HyperV | VMWareVC | VMWareESX | XENServer} ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: NewVMHostsTemplate

```
Get-SCVMHostRating -DiskSpaceGB <UInt16> -VMHost <Host[]> -VMName <String> -VMTemplate <Template> [-CPUExpectedUtilizationPercent <UInt16> ] [-CPUPriority <UInt16> ] [-DiscoveryID <Guid> ] [-DiskIOExpectedCountPerSecond <Int32> ] [-DiskPriority <UInt16> ] [-IsMigration] [-JobVariable <String> ] [-MemoryPriority <UInt16> ] [-NetworkPriority <UInt16> ] [-NetworkUtilizationExpectedMbps <Int32> ] [-PlacementGoal {LoadBalance | Consolidate} ] [-ReturnFirstSuitableHost] [-RunAsynchronously] [-UseDefaultPath] [-VirtualizationPlatform {Unknown | VirtualServer | HyperV | VMWareVC | VMWareESX | XENServer} ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

## Detailed Description

The Get-SCVMHostRating cmdlet calculates the placement rating for one or more hosts managed by Virtual Machine Manager (VMM) on which you might want to deploy a specific virtual machine.

The rating indicates the suitability of a computer to serve as a host for a virtual machine that requires a specific hardware configuration. The rating can be computed by individual host, for an array of hosts, or for each host that belongs to a specific host group or set of host groups.

When you run the Get-SCVMHostRating cmdlet, VMM returns an SCVMHostRating object for each of the specified hosts based on the hardware configuration that you want on the virtual machine. You can also specify additional placement options in order to modify how the ratings are calculated.

If you supply multiple host objects or an array of host objects to Get-SCVMHostRating, VMM gathers information about the host objects from the VMM database. To produce a host rating, VMM then compares the running state of the virtual machine against the database information. This operation does not guarantee migration compatibility of the virtual machine with a target host.

If you supply a single host object that is running Windows Server 2008 R2, or later, VMware, or Citrix XenServer to Get-VMHostRating, the cmdlet performs a direct validation of the running state of the virtual machine against the target host. Performing a direct validation ensures migration compatibility of

---

the virtual machine. When performing the direct validation, the command might take several seconds to complete.

## Parameters

### **-Cloud<Cloud[]>**

Specifies a private cloud object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-CPUExpectedUtilizationPercent<UInt16>**

Specifies the percent of CPU on the host that you expect this virtual machine to use. This value is used only when VMM determines a suitable host for the virtual machine.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-CPUPriority<UInt16>**

Specifies the relative importance of CPU utilization for a virtual machine on a host. To make CPU utilization a higher priority relative to other factors (such as disk I/O performance, memory utilization, and network utilization), set this value to a higher number. Valid values: 0 through 10. Default value: 5.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-DiscoveryID<Guid>**

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-DiskIOExpectedCountPerSecond<Int32>**

Specifies the number of disk input/output operations per second (IOPS) that you expect this virtual machine to use.

Example format: -DiskIO 1500 (to specify 1500 IOPS).

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-DiskPriority<UInt16>**

Specifies the relative importance of disk input/output (I/O) performance for a virtual machine on a host. To make disk I/O performance a higher priority relative to other factors (such as CPU utilization,

---

memory utilization, and network utilization), set this value to a higher number. Valid values: 0 through 10. Default value: 2.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-DiskSpaceGB<UInt16>**

Specifies, in gigabytes (GB), the amount of hard disk space on the host that can be used by a specific virtual machine.

Example: -DiskSpaceGB 20 (to specify 20 GB of disk space)

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-HardwareProfile<HardwareProfile>**

Specifies a hardware profile object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

---

## **-HighlyAvailable<Boolean>**

Specifies that a virtual machine will be placed on a Hyper-V host that is part of a host cluster. Configure this setting on a virtual machine, or on a template or hardware profile that will be used to create virtual machines.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-IsCloudOnlyRating**

Indicates that the rating only applies to a private cloud.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-IsMigration**

Specifies that a rating indicating a computer's suitability as a host to which to move a virtual machine will be calculated even if the source and destination host is the same computer.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false



---

### **-JobGroup<Guid>**

Specifies an identifier for a series of commands that will run as a set just before the final command that includes the same job group identifier runs.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-JobVariable<String>**

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-LUNCountRequirement<UInt32>**

Specifies the number of LUNs required by a virtual machine when evaluating which computers are suitable hosts on which to deploy this virtual machine.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false

---

Accept Wildcard Characters?	false
-----------------------------	-------

### **-MemoryPriority<UInt16>**

Specifies the relative importance of memory utilization by a virtual machine on a host. To make memory utilization a higher priority relative to other factors (such as CPU utilization, disk I/O performance, and network utilization), set this value to a higher number. Valid Values: 0 through 10. Default value: 8.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-NetworkPriority<UInt16>**

Specifies the relative importance of network utilization by a virtual machine on a host. To make network utilization a higher priority relative to other factors (such as CPU utilization, disk I/O performance, and memory utilization), set this value to a higher number. Valid values: 0 through 10. Default value: 2.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-NetworkUtilizationExpectedMbps<Int32>**

Specifies, in megabits per second (Mbps), the amount of traffic on the physical host's network that you expect this virtual machine to use.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-OperatingSystem<OperatingSystem>**

Specifies the type of operating system for a virtual machine. To list the names of all available operating systems in VMM, type: "Get-SCOperatingSystem".

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-PlacementGoal<EnginePlacementGoals>**

Specifies the placement algorithm to use when VMM selects the most suitable host on which to deploy a virtual machine. Load balancing among hosts lets VMM minimize the processing load on any one host. Consolidation lets VMM maximize resources by combining multiple low-utilization workloads on a single host. Valid values: LoadBalance, Consolidate.

The acceptable values for this parameter are:

LoadBalance	
Consolidate	

Aliases	none
Required?	false
Position?	named
Default Value	none

---

Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-ReturnFirstSuitableHost**

When used in conjunction with Get-SCVMHostRating or Update-SCServiceConfiguration, instructs the Intelligent Placement engine to return the first suitable host per virtual machine to be placed and stop processing Placement for said virtual machine.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-RunAsynchronously**

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-UseDefaultPath**

Specifies that only volumes for which a default path has been set on the host will be evaluated as possible candidates for virtual machine placement. If you omit this parameter or if no default paths are set on the host, all volumes will be evaluated by the placement process.

Aliases	none
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VirtualizationPlatform<VirtualizationPlatform>**

Specifies the virtualization platform of a virtual machine host managed by VMM. Valid values are: HyperV, VMWareESX, XENServer.

The acceptable values for this parameter are:

Unknown	
VirtualServer	
HyperV	
VMWareVC	
VMWareESX	
XENServer	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VM<VM>**

Specifies a virtual machine object.

Aliases	none
Required?	true
Position?	named

---

Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-VMConfiguration<BaseVMConfiguration>**

Specifies a virtual machine configuration object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-VMHost<Host[]>**

Specifies an array of virtual machine host objects.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-VMHostGroup<HostGroup[]>**

Specifies a virtual machine host group object or an array of host group objects.

Aliases	none
Required?	true
Position?	named

---

Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-VMMServer<ServerConnection>**

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-VMName<String>**

Specifies the name of a virtual machine to be placed on a physical host server. Use this parameter to verify that another virtual machine with the same name is not already deployed on that host.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMTemplate<Template>**

Specifies a VMM template object used to create virtual machines.

Aliases	none
Required?	true
Position?	named

---

Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **VMHostRating**

## Examples

### 1: Calculate host ratings for a specific server as a possible host for an existing virtual machine.

The first command gets the virtual machine object named VM01 and stores the object in the \$VM variable.

The second command gets the host object named VMHost02 and stores the object in the \$VMHost variable.

The third command returns the placement rating for VMHost02 that indicates its suitability as a host for VM01 and stores the rating information in the \$HostRating variable.

The last command displays the host ratings stored in \$HostRating to the user.

NOTE: Because the preceding example supplies a single host object to Get-SCVMHostRating, if the host is running Windows Server 2008 R2, or later, VMware, or XenServer, it will perform a direct validation of the running state of the virtual machine against the target host to ensure migration compatibility of the virtual machine.

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM01"
PS C:\> $VMHost = Get-SCVMHost -ComputerName "VMHost02.Contoso.com"
PS C:\> $HostRating = Get-SCVMHostRating -VM $VM -VMHost $VMHost
PS C:\> $HostRating
```

### 2: Calculate host ratings for each server in a host group as a possible host for an existing virtual machine.

The first command gets the virtual machine object named VM02 and stores the object in the \$VM variable.



---

The second command gets the host group object that named HostGroup02 and stores the object in the \$VMHostGroup variable.

The third command returns the placement ratings for all hosts in the specified host group and indicates the suitability of each host in that host group as a host for VM02. The command stores the rating information in \$HostRatings.

The last command displays the host ratings stored in \$HostRating to the user.

NOTE: Because the preceding example supplies multiple host objects to Get-SCVMHostRating, it will not perform a direct validation of the virtual machine against the hosts to produce host ratings. To ensure migration compatibility, you should perform a direct validation by running the Get-SCVMHostRating cmdlet on each potential target host individually.

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM02"
PS C:\> $VMHostGroup = Get-SCVMHostGroup -Name "HostGroup02"
PS C:\> $HostRatings = Get-SCVMHostRating -VM $VM -VMHostGroup $VMHostGroup
PS C:\> $HostRatings
```

### **3: Calculate host ratings for each server in a host group as a possible host for a new virtual machine.**

The first command gets the host group object named HostGroup03 and stores the object in the \$VMHostGroup variable.

The second command gets the hardware profile object named "HWProfile01" and stores the object in the \$HWProfile variable.

The third command returns the placement ratings for all hosts in the specified host group for a new virtual machine and stores the placement ratings in \$HostRatings. Before determining the host ratings, this command modifies the priorities for various factors by using the following parameters to specify these values: DiskSpaceGB, CPUPriority, MemoryPriority, DiskPriority, and NetworkPriority. See the individual parameter descriptions for additional information.

The last command displays the host ratings stored in \$HostRatings to the user.

NOTE: Because the preceding example supplies multiple host objects to Get-SCVMHostRating, it will not perform a direct validation of the virtual machine against the hosts to produce host ratings. To ensure migration compatibility, you should perform a direct validation by running the Get-SCVMHostRating cmdlet on each potential target host individually.

```
PS C:\> $VMHostGroup = Get-SCVMHostGroup -Name "HostGroup03"
PS C:\> $HWProfile = Get-SCHardwareProfile | where {$_.Name -eq "HWProfile01"}
PS C:\> $HostRatings = Get-SCVMHostRating -VMHostGroup $VMHostGroup -HardwareProfile
$HWProfile -DiskSpaceGB 20 -VMName "VM03" -CPUPriority 8 -MemoryPriority 5 -DiskPriority 3 -
NetworkPriority 1
PS C:\> $HostRatings
```

---

## 4: Calculate host ratings for each host in an array as a possible host for a new virtual machine.

The first command gets the operating system object that represents a 64-bit edition of Windows Server 2008 R2 Standard edition and stores the object in the \$OS variable.

The second command generates a GUID and stores the GUID in \$JobGroupID. The job group ID functions as an identifier that groups subsequent commands into a single job group.

The third command creates a new virtual disk drive with the specified properties, but uses the job group ID to specify that the virtual disk drive is not created until just before the Get-SCVMHostRating cmdlet in the last command runs.

The fourth and fifth commands retrieve an array of host objects and a specific hardware profile object to pass into the Get-VMHostRating cmdlet in the next command.

The sixth command returns the placement ratings for all hosts in the specified host list and indicates the suitability of each host in that list for the new virtual machine with the specified characteristics. The command stores the rating information in \$HostRatings.

Before the Get-SCVMHostRating cmdlet returns the host ratings, the command uses the JobGroup parameter to run the New-SCVirtualDiskDrive command from the third command so that the Get-SCVMHostRating cmdlet includes the virtual disk drive and its settings when calculating placement ratings.

The last command displays the host ratings stored in \$HostRatings to the user.

NOTE: Because the preceding example supplies an array of host objects to Get-SCVMHostRating, it will not perform a direct validation of the virtual machine against the hosts to produce host ratings. To ensure migration compatibility, you should perform a direct validation by running the Get-VMHostRating cmdlet on each potential target host individually.

```
PS C:\> $OS = Get-SCOperatingSystem | where {$_.Name -eq "64-bit edition of Windows Server 2008 R2 Standard"}
PS C:\> $JobGroupID = [guid]::NewGuid()
PS C:\> New-SCVirtualDiskDrive -SCSI -Fixed -Bus 0 -Lun 2 -Size 10 -JobGroup $JobGroupID -FileName "TestDiskDrive"
PS C:\> $VMHosts = Get-SCVMHost
PS C:\> $HWProfile = Get-SCHardwareProfile | where { $_.Name -eq "HWProfile01" }
PS C:\> $HostRatings = Get-SCVMHostRating -DiskSpaceGB 10 -HardwareProfile $HWProfile -VMHost $VMHosts -VMName "VM04" -OperatingSystem $OS -JobGroup $JobGroupID
PS C:\> $HostRatings
```

## 5: Calculate host ratings for a specific VMM management server as a possible host for an existing virtual machine.

The first command gets the virtual machine object named VM05 and stores the object in the \$VM variable.

The second command gets the host object named VMHost05 and stores the object in the \$VMHost variable.

---

The third command returns the placement rating for VMHost05 which indicates its suitability as a host for VM05 based on a particular set of customized priority ratings and based on consolidation as the placement goal (as opposed to the default, load balancing). The command stores the rating information in \$HostRating.

The last command displays the host rating stored in \$HostRating to the user.

NOTE: Because the preceding example supplies a single host object to Get-VMHostRating, if the host is running Windows Server 2008 R2, or later, VMware, or XenServer, it will perform a direct validation of the running state of the virtual machine against the target host to ensure migration compatibility of the virtual machine.

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM05"
PS C:\> $VMHost = Get-SCVMHost -ComputerName "VMHost05.Contoso.com"
PS C:\> $HostRating = Get-SCVMHostRating -VM $VM -VMHost $VMHost -CPUPriority 6 -
DiskPriority 5 -MemoryPriority 4 -NetworkPriority 4 -PlacementGoal "Consolidate"
PS C:\> $HostRating
```

## 6: Calculate host ratings for a new virtual machine based on a specific virtual machine template.

The first command gets the virtual machine template object named "VMTemplate01" and stores the object in the \$VMTemplate variable.

The second command gets the host object named VMHost01 and stores the object in the \$VMHost variable.

The third command returns the placement ratings for a new virtual machine if it were created by using VMTemplate01 and if it were to be placed on host VMHost01. The command stores the ratings in \$HostRating.

The last command displays the host ratings stored in \$HostRating to the user.

NOTE: The DiskSpaceGB parameter is required even though the template might already have a virtual hard disk with a specified amount of disk space. Requiring the DiskSpaceGB parameter ensures that a certain minimum amount of hard disk space is available on the host that can be used by the virtual machine. If the amount of space specified for the virtual hard disk in the template is larger than the size specified by using the DiskSpaceGB parameter, the larger of the two sizes is taken into consideration when computing the host ratings.

NOTE: Because the preceding example supplies a single host object to Get-VMHostRating, if the host is running Windows Server 2008 R2, or later, VMware, or XenServer, it will perform a direct validation of the running state of the virtual machine against the target host to ensure migration compatibility of the virtual machine.

```
PS C:\> $VMTemplate = Get-SCVMTemplate | where {$_.Name -eq "VMTemplate01"}
PS C:\> $VMHost = Get-SCVMHost -ComputerName "VMHost01.Contoso.com"
PS C:\> $HostRating = Get-SCVMHostRating -DiskSpaceGB 5 -VMTemplate $VMTemplate -VMHost
$VMHost -VMName "VM06"
PS C:\> $HostRating
```

---

## 7: Calculate host ratings for a specific host as a possible host for all virtual machines.

The first command gets the host object that named VMHost02 and stores the host object in the \$VMHost variable.

The second command gets all virtual machines objects in your environment and saves these objects in the \$VMs object array. If your environment has a very large number of virtual machines, you might want to use a filter to select a subset of virtual machines.

The third command returns the placement ratings for VMHost02 which indicate its suitability as a host for each of the virtual machine objects in \$VMs and stores the rating information in \$RatingArray. For more information about the Windows PowerShell foreach loop statement, type: Get-Help about\_Foreach.

The last command displays the ratings stored in \$RatingArray for the user.

NOTE: This example computes the ratings for each virtual machine individually on a host. If you want to place multiple virtual machines on a host, create a temporary hardware profile with the aggregated resource demands and pass it to Get-SCVMHostRating.

NOTE: Because the preceding example supplies a single host object to Get-VMHostRating, if the host is running Windows Server 2008 R2, or later, VMware, or XenServer, it will perform a direct validation of the running state of the virtual machine against the target host to ensure migration compatibility of the virtual machine.

```
PS C:\> $VMHost = Get-SCVMHost -ComputerName "VMHost02.Contoso.com"
```

```
PS C:\> $VMs = Get-SCVirtualMachine
```

```
PS C:\> $RatingArray = @(ForEach ($VM in $VMs) {Get-VMHostRating -VM $VM -VMHost $VMHost} )
```

```
PS C:\> $RatingArray
```

### Related topics

[Get-SCHardwareProfile](#)

[Get-SCVirtualMachine](#)

[Get-SCVMHost](#)

[Get-SCVMHostGroup](#)

---

# Get-SCVMMAccessLicense

---

## Get-SCVMMAccessLicense

Gets the VMM server license information.

### Syntax

Parameter Set: License

```
Get-SCVMMAccessLicense -License -VMMServer <ServerConnection> [ <CommonParameters>]
```

Parameter Set: LicenseBy

```
Get-SCVMMAccessLicense -LicenseBy {ManagementServer | SML | VOSE} -VMMServer  
<ServerConnection> [ <CommonParameters>]
```

### Detailed Description

The Get-SCVMMAccessLicense cmdlet gets the Virtual Machine Manager (VMM) server license information.

### Parameters

#### -License

Indicates that all available licenses are returned.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

#### -LicenseBy<LicenseBy>

Specifies how VMM is licensed. Valid values are: SML, ManagementServer, VOSE.

---

The acceptable values for this parameter are:

ManagementServer	
SML	
VOSE	

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Examples**

### **1: Get a list of available licenses for VMM.**

This command returns a list of available licenses for the VMM server VMMServer01.

---

```
PS C:\> Get-SCVMMAccessLicense -VMMServer "VMMServer01.Contoso.com" -License
```

## **2: Get all licenses that have a LicenseBy value of "ManagementServer".**

This command gets all management server licenses on VMM server VMMServer01.

```
PS C:\> Get-SCVMMAccessLicense -VMMServer "VMMServer01.Contoso.com" -LicenseBy "ManagementServer"
```

## **3: Get all licenses that have a LicenseBy value of "SML".**

This command returns a list of licensed hosts associated with VMM server VMMServer01.

```
PS C:\> Get-SCVMMAccessLicense -VMMServer "VMMServer01.Contoso.com" -LicenseBy "SML"
```

## **4: Get all licenses that have a LicenseBy value of "VOSE".**

Returns a list of licensed virtual machines associated with VMM server VMMServer01.

```
PS C:\> Get-SCVMMAccessLicense -VMMServer "VMMServer01.Contoso.com" -LicenseBy "VOSE"
```

## **Related topics**

[Register-SCVMMAccessLicense](#)

---

# Get-SCVMMDRServiceRegistration

---

## Get-SCVMMDRServiceRegistration

This cmdlet is reserved for future use.

### Syntax

Parameter Set: Default

```
Get-SCVMMDRServiceRegistration [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

This cmdlet is reserved for future use.

### Parameters

#### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

#### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

### Outputs

The output type is the type of the objects that the cmdlet emits.



- 
- `DRVMMRegistration`

## Examples

**PS C:\> # The Get-SCVMMDRServiceRegistration cmdlet is reserved for future use.**

## Related topics

[Get-SCVMMServer](#)

---

# Get-SCVMMManagedComputer

---

## Get-SCVMMManagedComputer

Gets computer objects managed by VMM.

### Syntax

Parameter Set: Default

```
Get-SCVMMManagedComputer [[-ComputerName] <String> ] [-VMMServer <ServerConnection> ] [ <CommonParameters> ]
```

### Detailed Description

The Get-SCVMMManagedComputer cmdlet gets one or more computer objects managed by Virtual Machine Manager (VMM). Managed computers include the following types of computers:

- VIRTUAL MACHINE HOST. A Hyper-V host, VMware ESX host, or Citrix XenServer host on which you deploy virtual machines.
- VMWARE VIRTUALIZATION MANAGER. A server running VMware vCenter Server that VMM connects to in order to manage ESX hosts and the virtual machines deployed on those hosts.
- LIBRARY SERVER. A server used to make shares available to store VMM library resources.
- P2V SOURCE COMPUTER. Any physical computer that you want to "clone" so that you can use its hardware and software settings to create one or more virtual machines.

### Parameters

#### **-ComputerName<String>**

Specifies the name of a computer that VMM can uniquely identify on your network. Valid formats are: FQDN, IPv4 or IPv6 address, or NetBIOS name.

NOTE: See the examples for a specific cmdlet to determine how that cmdlet specifies the computer name.

Aliases	none
---------	------

Required?	false
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **VMMManagedComputer**

## **Examples**

### **1: Get all computers managed by VMM.**

This command gets all computer objects managed by VMM and displays information about these managed computers to the user. When you look at the output, note that the RoleString property indicates whether the server is a library server, a host for virtual machines, both a library server and a host, a VMware vCenter Server, or a Citrix XenServer.

```
PS C:\> Get-SCVMMManagedComputer
```

---

## 2: Update the agent software on all host servers managed by VMM.

The first command uses `Get-Credential` to prompt you to supply a user name and password and then stores your credentials in the `$Credential` variable. The required credentials for this operation is a domain account with rights to update software on computers managed by VMM.

The second command gets all computer objects managed by VMM, and then passes each managed computer object to the `Update-SCVMMManagedComputer` cmdlet which updates the VMM agent software on each computer. As this command is processed, `$Credential` provides credentials to the `Update-SCVMMManagedComputer` cmdlet.

```
PS C:\> $Credential = Get-Credential
```

```
PS C:\> Get-SCVMMManagedComputer | Update-SCVMMManagedcomputer -Credential $Credential -RunAsynchronously
```

## 3: Get a specific computer managed by VMM by IP address.

This command gets a computer object by its IP address.

```
PS C:\> Get-SCVMMManagedComputer -ComputerName "10.20.30.40"
```

## Related topics

[Get-SCLibraryServer](#)

[Get-SCVMHost](#)

[New-SCP2V](#)

[Register-SCVMMManagedComputer](#)

[Update-SCVMMManagedComputer](#)

---

# Get-SCVMMServer

---

## Get-SCVMMServer

Connects to a VMM management server if a connection does not already exist, or connects to a different VMM management server.

### Syntax

Parameter Set: Default

```
Get-SCVMMServer [-ComputerName] <String> [-AllowJobGC] [-ConnectAs {Administrator |  
DelegatedAdmin | TenantAdmin | SelfServiceUser | ReadOnlyAdmin} ] [-Credential  
<PSCredential> ] [-ForOnBehalfOf] [-RetainDeletedObjects] [-RetainObjectCache] [-  
SetAsDefault] [-TCPPort <Int32> ] [-UserRoleName <String> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCVMMServer cmdlet connects to a Virtual Machine Manager (VMM) management server if a connection does not already exist and retrieves the server object from the VMM database. The default port used to connect to a VMM server is TCP port 8100.

If you have created a connection to a VMM management server using the user interface (UI), when you open a VMM command shell, that command shell automatically connects to the same VMM management server. If you have not previously connected to the VMM management server using the UI, you need to use Get-SCVMMServer to establish a connection. Note that if you connect to a VMM management server only using the VMM command shell, you will need to use the SetAsDefault parameter to retain the connection from session to session, or re-connect to the VMM server each time you open a new VMM command shell session.

You can also use Get-VMMServer to connect to a different VMM management server.

After a connection to the VMM management server is established, all future commands run at the VMM command shell command line that require the VMM server object will automatically use the existing connection until you close that VMM command shell window.

The VMM service running on the VMM management server supports the VMM database. This database is stored in Microsoft SQL Server either on the VMM management server itself or on a separate server running SQL Server.

The VMM service enables you to manage your virtual environment, including host servers (which host virtual machines), library servers (which store VMM library resources), and virtual machines deployed on a host or stored in the library.

For more information about connecting to the VMM management server, type: "Get-Help about\_VMM\_2012\_Connecting\_to\_the\_VMM\_Server".

---

## Parameters

### **-AllowJobGC**

Indicates that garbage collection is allowed for jobs.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-ComputerName<String>**

Specifies the name of a computer that VMM can uniquely identify on your network. Valid formats are: FQDN, IPv4 or IPv6 address, or NetBIOS name.

NOTE: See the examples for a specific cmdlet to determine how that cmdlet specifies the computer name.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-ConnectAs<Profile>**

Specifies the VMM user role to use, if the user is a member of more than one role, when connecting to the VMM management server from the VMM command shell. Valid values are: Administrator, DelegatedAdmin, ReadOnlyAdmin, SelfServiceUser.

VMM Administrators can manage all VMM objects. Delegated administrators and self-service users can access and change only the objects that are within the scope of their user roles. Read-Only administrators can only view the properties of existing objects; they cannot create new objects or change the properties of existing objects.

The acceptable values for this parameter are:

Administrator	
DelegatedAdmin	
TenantAdmin	
SelfServiceUser	
ReadOnlyAdmin	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Credential<PSCredential>**

Specifies a credential object or, for some cmdlets, a Run As account object that contains the user name and password of an account that has permission to perform this action. Or, in the case of Restart-SCJob, has permission to complete a restarted task.

For more information about the PSCredential object, type: "Get-Help Get-Credential".

For more information about Run As accounts, type: "Get-Help New-SCRunAsAccount".

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-ForOnBehalfOf**

Indicates that an administrator is able to use "OnBehalfOf" parameters, when available, for VMM cmdlets. This parameter is primarily used for Service Provider Foundation connections to VMM.

---

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-RetainDeletedObjects**

Specifies that objects in the cache that are marked for deletion will be preserved. You might need this parameter only if you create a user interface on top of the VMM command shell.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-RetainObjectCache**

Specifies that the objects in the cache will remain in memory and will not be reclaimed by garbage collection. You might need this parameter if you create a user interface on top of the VMM command shell.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false



---

## **-SetAsDefault**

Indicates, when set to \$True, that the VMM command shell connects to the specified VMM management server for this session and retains that connection for future sessions.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-TCPPort<Int32>**

Specifies a numeric value that represents a TCP port.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-UserRoleName<String>**

Specifies the name of a user role. Types of user roles that are named include Delegated Administrator, Read-Only Administrator and Self-Service User.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## Examples

### 1: Connect to a VMM server.

This command connects to the VMM server named VMMServer01 located in the Contoso domain and gets the server object from the VMM database.

```
PS C:\> Get-SCVMMServer -ComputerName "VMMServer01.Contoso.com"
```

### 2: Connect to a VMM server through a specific port.

This command connects over TCP port 8100 to the VMM server named VMMServer01 located in the Contoso domain.

```
PS C:\> Get-SCVMMServer -ComputerName "VMMServer01.Contoso.com" -TCPPort 8100
```

### 3: Connect to a VMM server and get its .NET object type, methods, and properties.

The first command gets the VMM server object named VMMServer01 and stores the object in the \$VMMServer variable.

The second command passes the VMM server object stored in \$VMMServer to the Get-Member cmdlet, which retrieves and displays the following:

- TypeName: The name of the .NET type of the VMM server object:

```
Microsoft.SystemCenter.VirtualMachineManager.Remoting.ServerConnection
```

- MemberType: A list containing the name and definition for each event, method, and property associated with this object type.

The third command retrieves and displays the same information as the second command, except that it pipes the output to the Format-List cmdlet so that the complete definition for each method and each property for the VMM server object is displayed.

```
PS C:\> $VMMServer = Get-SCVMMServer -ComputerName "VMMServer01.Contoso.com"
```

```
PS C:\> $VMMServer | Get-Member
```

```
PS C:\> $VMMServer | Get-Member | Format-List
```

---

## 4: Connect to a different VMM server with a different user role.

This command connects to the VMM server named VMMServer02 located in the Contoso domain using the DelegatedAdmin user role.

```
PS C:\> Get-SCVMMServer -ComputerName "VMMServer02.Contoso.com" -ConnectAs "DelegatedAdmin"
```

### Related topics

[Set-SCVMMServer](#)

---

# Get-SCVMNetwork

---

## Get-SCVMNetwork

Gets a VM network.

### Syntax

Parameter Set: All

```
Get-SCVMNetwork [[-Name] <String> ] [-LogicalNetwork <LogicalNetwork> ] [-Unassigned] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ByID

```
Get-SCVMNetwork [-ID] <Guid> [[-Name] <String> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCVMNetwork cmdlet gets a VM network.

### Parameters

#### **-ID<Guid>**

Specifies the numerical identifier (as a globally unique identifier, or GUID) for a specific object.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

---

## **-LogicalNetwork<LogicalNetwork>**

Specifies a logical network. A logical network is a named grouping of IP subnets and VLANs that is used to organize and simplify network assignments.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	false
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Unassigned**

Retrieves IP addresses or MAC addresses that have been allocated from an address pool but not assigned to a resource.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Examples**

### **1: Get a VM network**

This command gets a VM network by its name.

```
PS C:\> $VMNet = Get-SCVMNetwork -Name "VMNetwork01"
```

## **Related topics**

[New-SCVMNetwork](#)

[Remove-SCVMNetwork](#)

[Set-SCVMNetwork](#)

---

# Get-SCVMNetworkGateway

---

## Get-SCVMNetworkGateway

Gets a VM network gateway object.

### Syntax

Parameter Set: ByVMNetwork

```
Get-SCVMNetworkGateway -VMNetwork <VMNetwork> [-VMMServer <ServerConnection> ] [ <CommonParameters> ]
```

Parameter Set: ByID

```
Get-SCVMNetworkGateway -ID <Guid> [-VMMServer <ServerConnection> ] [ <CommonParameters> ]
```

### Detailed Description

The Get-SCVMNetworkGateway cmdlet gets one or more VM network gateway objects.

### Parameters

#### **-ID<Guid>**

Specifies the numerical identifier (as a globally unique identifier, or GUID) for a specific object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

#### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

---

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **-VMNetwork<VMNetwork>**

Specifies a VM network object.

To get a VM network object, use the Get-SCVMNetwork cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## **Related topics**

[Get-SCVMNetwork](#)

[Add-SCVMNetworkGateway](#)

[Read-SCVMNetworkGateway](#)

[Remove-SCVMNetworkGateway](#)

[Set-SCVMNetworkGateway](#)



---

# Get-SCVMSubnet

---

## Get-SCVMSubnet

Gets a VM subnet.

### Syntax

Parameter Set: All

```
Get-SCVMSubnet [[-Name] <String> ] [-Subnet <String> ] [-Unassigned] [-VMMServer  
<ServerConnection> ] [-VMNetwork <VMNetwork> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCVMSubnet cmdlet gets a VM subnet object.

### Parameters

#### **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	false
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

#### **-Subnet<String>**

Specifies an IP subnet (IPv4 or IPv6) in Classless Inter-Domain Routing (CIDR) notation.

Example format for an IPv4 subnet: 192.168.0.1/24

Example format for an IPv6 subnet: FD4A:29CD:184F:3A2C::/64

NOTE: An IP subnet cannot overlap with any other subnet in a host group or child host groups.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Unassigned**

Retrieves IP addresses or MAC addresses that have been allocated from an address pool but not assigned to a resource.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

---

## **-VMNetwork<VMNetwork>**

Specifies a VM network object.

To get a VM network object, use the Get-SCVMNetwork cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Examples**

### **1: Get a VM subnet by its name.**

This command gets the VM subnet object named VMSubnet01 and stores the object in the \$VMSubnet variable.

```
PS C:\> $VMSubNet = Get-SCVMSubnet -Name "VMSubnet01"
```

## **Related topics**

[New-SCVMSubnet](#)

[Remove-SCVMSubnet](#)

[Set-SCVMSubnet](#)

---

# Get-SCVMTemplate

---

## Get-SCVMTemplate

Gets virtual machine template objects from the VMM library.

### Syntax

Parameter Set: All

```
Get-SCVMTemplate [-All] [-ComputerTierTemplate <ComputerTierTemplate> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ID

```
Get-SCVMTemplate [-ComputerTierTemplate <ComputerTierTemplate> ] [-ID <Guid> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: Name

```
Get-SCVMTemplate [-ComputerTierTemplate <ComputerTierTemplate> ] [-Name <String> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCVMTemplate cmdlet gets virtual machine template objects from the Virtual Machine Manager (VMM) library

For information about how virtual machine templates are used to create new virtual machines, type: "Get-Help New-Template -detailed".

### Parameters

#### -All

Retrieves a full list of all subordinate objects independent of the parent object. For example, the command Get-SCVirtualDiskDrive -All retrieves all virtual disk drive objects regardless of the virtual machine object or template object that each virtual disk drive object is associated with.

Aliases	none
Required?	false
Position?	named
Default Value	none

---

Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-ComputerTierTemplate<ComputerTierTemplate>**

Specifies a computer tier template object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-ID<Guid>**

Specifies the numerical identifier (as a globally unique identifier, or GUID) for a specific object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **Template**

## **Examples**

### **1: Get all templates stored in the library.**

This command gets all template objects from the VMM library on VMMServer01, and then displays information about these templates to the user.

```
PS C:\> Get-SCVMTemplate -VMMServer "VMMServer01.Contoso.com"
```

### **2: Get all templates stored in the library that have a similar name.**

This command gets all template objects from the VMM library on VMMServer01 whose name begins with "Windows Server 2008", and then displays information about these templates to the user.

```
PS C:\> Get-SCVMTemplate -VMMServer "VMMServer01.Contoso.com" | where { $_.Name -like "Windows Server 2008*" }
```

---

## Related topics

[New-SCVMTemplate](#)

[Remove-SCVMTemplate](#)

[Set-SCVMTemplate](#)

---

# Get-SCVMwareResourcePool

---

## Get-SCVMwareResourcePool

Gets VMware resource pool objects from the VMM database.

### Syntax

Parameter Set: FilterById

```
Get-SCVMwareResourcePool -ID <Guid> [ <CommonParameters>]
```

Parameter Set: FilterByVMHost

```
Get-SCVMwareResourcePool -VMHost <Host> [ <CommonParameters>]
```

Parameter Set: FilterByVMHostCluster

```
Get-SCVMwareResourcePool -VMHostCluster <HostCluster> [ <CommonParameters>]
```

### Detailed Description

The Get-SCVMWareResourcePool cmdlet gets VMware resource pool objects from the Virtual Machine Manager (VMM) database.

VMware resource pools are imported when a ESX host or ESX host cluster that owns the resource pools are added to VMM.

VMware uses resource pools to group virtual machines deployed on ESX hosts, or ESX host clusters, into an organizational hierarchy that consists of parent, sibling, and child resource pools. Resources, such as CPU and memory, are specified for virtual machines assigned to each resource pool.

Administration of sets of resource pools can be delegated, in vCenter Server, to administrators by department, by geographical region, or by some other organizational requirement.

VMware resource pools can provide resources to private clouds. For more information about creating a private cloud, type: "Get-Help New-SCCloud".

### Parameters

#### -ID<Guid>

Specifies the numerical identifier (as a globally unique identifier, or GUID) for a specific object.

Aliases	none
Required?	true



Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMHost<Host>**

Specifies a virtual machine host object. VMM supports Hyper-V hosts, VMware ESX hosts, and Citrix XenServer hosts.

For more information about each type of host, type: "Get-Help Add-SCVMHost -detailed". See the examples for a specific cmdlet to determine how that cmdlet uses this parameter.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-VMHostCluster<HostCluster>**

Specifies a VMM host cluster object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

---

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **VMwareResourcePool**

## Examples

### 1: Get the VMware resource pool for a VMware ESX host.

The first command gets the object that represents the VMware ESX host named VMHost01 from the VMM database by specifying the IP address (represented in the example by "nnn.nnn.nnn.nnn") of the ESX host. This ESX Server is managed by VMM through VMware VirtualCenter Server. The command stores the host object in variable \$MyESXHost.

The last command gets the VMware resource pool information from the VMM database for the ESX host object stored in variable \$VMHost and displays information about the resource pool to the user.

```
PS C:\> $ESXHost = Get-SCVMHost -ComputerName "ESXHost01.Contoso.com"
```

```
PS C:\> Get-SCVMwareResourcePool -VMHost $ESXHost
```

## Related topics

[Get-SCVMHost](#)

[Get-SCVMHostCluster](#)

---

# Get-SCVMXComputerConfiguration

---

## Get-SCVMXComputerConfiguration

Gets VMX computer configuration objects from the VMM database that are associated with one or more VMware-based virtual machines.

### Syntax

Parameter Set: Default

```
Get-SCVMXComputerConfiguration [-VMMServer <ServerConnection> ] [-VMXPath <String> ] [ <CommonParameters> ]
```

### Detailed Description

The Get-SCVMXComputerConfiguration cmdlet gets one or more VMX computer configuration objects from the Virtual Machine Manager (VMM) database that are associated with one or more VMware virtual machines. Information about a virtual machine computer's hardware, disks, and operating system is stored in the VMX computer configuration object.

A VMX computer configuration object is used by the New-SCV2V cmdlet when it converts a VMware-based virtual machine deployed on an ESX host to a virtual machine deployed on a Windows-based Hyper-V host.

### Parameters

#### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

---

## **-VMXPath<String>**

Specifies the full UNC path to the .vmx file of a VMware virtual machine.

Example format: \\ServerName\VolumeName\DirectoryName\VMwareVM.vmx

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **VMXMachineConfig**

## **Examples**

### **1: Get all VMX computer configuration objects in your VMM environment.**

This command gets all VMX computer configuration objects from the VMM database and displays information about these VMX computer configuration objects to the user.

```
PS C:\> Get-SCVMXComputerConfiguration -VMMServer "VMMServer01.Contoso.com"
```

## **Related topics**

[New-SCV2V](#)

[New-SCVMXComputerConfiguration](#)

[Remove-SCVMXComputerConfiguration](#)

---

# Get-SCVPNConnection

---

## Get-SCVPNConnection

Gets a VPN connection.

### Syntax

Parameter Set: ByVMNetworkGateway

```
Get-SCVPNConnection -VMNetworkGateway <VMNetworkGateway> [-VMMServer <ServerConnection> ] [ <CommonParameters> ]
```

Parameter Set: ByID

```
Get-SCVPNConnection -ID <Guid> [-VMMServer <ServerConnection> ] [ <CommonParameters> ]
```

### Detailed Description

The Get-SCVPNConnection cmdlet gets a virtual private network (VPN) connection.

### Parameters

#### **-ID<Guid>**

Specifies the numerical identifier (as a globally unique identifier, or GUID) for a specific object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

#### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **-VMNetworkGateway<VMNetworkGateway>**

Specifies a VM network gateway object.

To obtain a VM network gateway object, use the Get-SCVMNetworkGateway cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## **Related topics**

[Add-SCVPNConnection](#)

[Remove-SCVPNConnection](#)

[Set-SCVPNConnection](#)

---

# Grant-SCIPAddress

---

## Grant-SCIPAddress

Allocates a static or virtual IP address from a specified address pool.

### Syntax

Parameter Set: Default

```
Grant-SCIPAddress -GrantToObjectType {VirtualNetworkAdapter | VIP | HostNetworkAdapter | LoadBalancerConfiguration | VirtualMachine | HostCluster | VMSubnet | NetworkGateway} - StaticIPAddressPool <StaticIPAddressPool> [-Description <String> ] [-GrantToObjectID <Guid> ] [-IPAddress <String> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Grant-SCIPAddress cmdlet allocates static IP and virtual IP addresses from a specified address pool.

To grant a specific IP address, use the IPAddress parameter. Otherwise, VMM will choose the IP address from the address pool.

### Parameters

#### **-Description<String>**

States a description for the specified object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-GrantToObjectID<Guid>**

Specifies the ID of an object to which an allocated IP address or MAC address will be assigned.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-GrantToObjectType<AllocatedToObjectType>**

Specifies a value for AllocatedToObjectType to which an allocated IP address or virtual IP address will be assigned. Valid values are: VirtualNetworkAdapter, VIP, HostNetworkAdapter, LoadBalancerConfiguration, VirtualMachine, HostCluster

The acceptable values for this parameter are:

VirtualNetworkAdapter	
VIP	
HostNetworkAdapter	
LoadBalancerConfiguration	
VirtualMachine	
HostCluster	
VMSubnet	
NetworkGateway	

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false



---

## **-IPAddress<String>**

Specifies an IPv4 or IPv6 address.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-StaticIPAddressPool<StaticIPAddressPool>**

Specifies an IP address pool from which you can assign static IP addresses.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **CloudDRPairingInfoData**

## Notes

- This cmdlet requires a VMM static IP address pool object, which can be retrieved using the Get-SCStaticIPAddressPool cmdlet.

## Examples

### 1: Allocate an IP address.

The first command gets the virtual machine object named VM01 and stores the object in the \$VM variable.

The second command gets the virtual network adapter objects for virtual machine VM01 and stores the objects in the \$vNICs variable.

The third command gets the static IP address pool object named Production IP Address Pool and stores the object in the \$IPPool variable.

The last command allocates an IP address from the static IP address pool stored in \$IPPool to the first virtual network adapter stored in \$vNICs, and supplies VM01 as the description for the allocated IP address.

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM01"
PS C:\> $vNICs = $VM.VirtualNetworkAdapters
PS C:\> $IPPool = Get-SCStaticIPAddressPool -Name "Production IP Address Pool"
PS C:\> Grant-SCIPAddress -StaticIPAddressPool $IPPool -GrantToObjectType
VirtualNetworkAdapter -GrantToObjectID $vNICs[0].ID -Description $VM.Name
```

## Related topics

[Get-SCIPAddress](#)

[Get-SCStaticIPAddressPool](#)

[Revoke-SCIPAddress](#)

[Set-SCIPAddress](#)

---

# Grant-SCMACAddress

---

## Grant-SCMACAddress

Allocates the next available physical address (MAC address) from a MAC address pool, and assigns it to a virtual network adapter.

### Syntax

Parameter Set: Default

```
Grant-SCMACAddress -MACAddressPool <MACAddressPool> -VirtualNetworkAdapter  
<VirtualNetworkAdapter> [-Description <String> ] [-JobVariable <String> ] [-MACAddress  
<String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [-  
Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The Grant-SCMACAddress cmdlet allocates the next available physical address (MAC address) from a MAC address pool and assigns it to a virtual network adapter. To allocate a specific MAC address, use the MACAddress parameter.

For information about creating MAC address pools, type: "New-SCMACAddressPool -detailed".

### Parameters

#### **-Description<String>**

States a description for the specified object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-MACAddress<String>**

Specifies the Media Access Control (MAC) address, or a set of MAC addresses, for a physical or virtual network adapter on a computer. Valid values are: one or more MAC addresses.

Example format for a single MAC address:

-MACAddress "00-15-5D-B4-DC-00"

Example format for a set of MAC addresses:

-MACAddress "00-15-5D-B4-DC-00", "00-1A-A0-E3-75-29"

Example format for a set of MAC addresses:

\$Macs = "00-15-5D-B4-DC-00", "00-1A-A0-E3-75-29"

Set-SCPXEserver -MACAddress \$Macs

NOTE: When used with New-SCPXEserver or Set-SCPXEserver, the MACAddress parameter updates the PXE interfaces from which the SCDM PXE Server listens for and responds to PXE requests

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-MACAddressPool<MACAddressPool>**

Specifies a MAC address pool.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VirtualNetworkAdapter<VirtualNetworkAdapter>**

Specifies a virtual network adapter object for a virtual machine.

TYPE OF HOST      NUMBER OF VIRTUAL NETWORK ADAPTERS

---

-----

Hyper-V Up to 4 emulated adapters per virtual machine.

Up to 8 synthetic adapters per virtual machine.

(Exception: no driver available for an emulated network adapter on a Windows Server 2003 x64 guest.)

VMware ESX Up to 4 emulated adapters per virtual machine.

Citrix XenServer Up to 7 emulated adapters per virtual machine.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-Confirm**

Prompts you for confirmation before executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false

---

Accept Wildcard Characters?	false
-----------------------------	-------

## -WhatIf

Describes what would happen if you executed the command without actually executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **CloudPairing[]**

## Notes

- Requires a VMM MACAddressPool object, which can be retrieved using the Get-SCMACAddressPool cmdlet, and a VMM virtual network adapter object, which can be retrieved using the Get-SCVirtualNetworkAdapter cmdlet.

## Examples

### 1: Allocate a MAC address from a MAC address Pool and assign it to a virtual network adapter.

The first command gets the virtual machine object named VM01 on host VMHost01 and stores the object in the \$VM variable.

The second command gets the virtual network adapter object for the virtual machine stored in \$VM and stores the object in the \$VNIC variable. This example assumes that the virtual machine has only one virtual network adapter.



---

The third command gets the host group object at the path All Hosts\HostGroup02\Production and stores the object in the \$HostGroup variable.

The fourth command gets the MAC address pool associated with the host group stored in \$HostGroup and named MAC Address Pool 01.

The last command assigns a MAC address to the virtual network adapter stored in \$VNIC.

```
PS C:\> $VM = Get-SCVirtualMachine -VMHost "VMHost01.Contoso.com" -Name "VM01"
```

```
PS C:\> $VNIC = Get-SCVirtualNetworkAdapter -VM $VM
```

```
PS C:\> $HostGroup = Get-SCVMHostGroup | where { $_.Path -eq "All  
Hosts\HostGroup02\Production" }
```

```
PS C:\> $MACAddressPool = Get-SCMACAddressPool -VMHostGroup $HostGroup -Name "MAC Address  
Pool 01"
```

```
PS C:\> Grant-SCMACAddress -MACAddressPool $MACAddressPool -VirtualNetworkAdapter $VNIC
```

## 2: Allocate a MAC address from a MAC address Pool and assign it to a specific virtual network adapter.

The first command gets the virtual machine object named VM02 and stores the object in the \$VM variable.

The second command gets the virtual network adapter object for VM02 with the slot ID of 1 and stores the object in the \$VNIC variable.

The third command gets the MAC address Pool named MAC Address Pool 02 and stores the object in the the \$MACAddressPool variable.

The last command grants a MAC address from MAC Address Pool 02 to the virtual network adapter stored in \$VNIC.

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM02"
```

```
PS C:\> $VNIC = Get-SCVirtualNetworkAdapter -VM $VM | where { $_.SlotId -eq 1 }
```

```
PS C:\> $MACAddressPool = Get-SCMACAddressPool -Name "MAC Address Pool 02"
```

```
PS C:\> Grant-SCMACAddress -MACAddressPool $MACAddressPool -VirtualNetworkAdapter $VNIC
```

## Related topics

[Get-SCMACAddress](#)

[Get-SCMACAddressPool](#)

[Revoke-SCMACAddress](#)

---

# Grant-SCResource

---

## Grant-SCResource

Grants a user or self-service user role access to a resource.

### Syntax

Parameter Set: Default

```
Grant-SCResource -Resource <ClientObject> [-JobGroup <Guid> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-UserName <String> ] [-UserRoleID <Guid[]> ] [-UserRoleName <String[]> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Grant-SCResource cmdlet grants a user or self-service user role access to a resource.

Types of resources that can be shared using Grant-SCResource include the following:

- Service Templates
- Virtual Machine Templates
- Guest Operating System Profiles
- SQL Server Profiles
- Hardware Profiles
- Application Profiles
- Services
- Virtual Machines

### Parameters

#### **-JobGroup<Guid>**

Specifies an identifier for a series of commands that will run as a set just before the final command that includes the same job group identifier runs.

Aliases	none
Required?	false
Position?	named

---

Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Resource<ClientObject>**

Specifies a resource object.

Aliases	none
Required?	true
Position?	named

---

Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-UserName<String>**

Specifies a the name of a user. Enter a user name with the format Domain\User.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-UserRoleID<Guid[]>**

Specifies the ID of a user role.

Aliases	none
Required?	false
Position?	named

Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-UserRoleName<String[]>**

Specifies the name of a user role. Types of user roles that are named include Delegated Administrator, Read-Only Administrator and Self-Service User.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

### **Outputs**

The output type is the type of the objects that the cmdlet emits.

- 
- Resource

## Examples

### 1: Share a resource with a specific user.

The first command gets the template object named Template01 and stores the object in the \$Resource variable.

The second command shares the resource stored in \$Resource (Template01) with the user named Katarina. If the user is a member of multiple self-service user roles with receive permission, then a user role must be specified.

```
PS C:\> $Resource = Get-SCVMTemplate | where {$_.Name -eq "Template01"}
PS C:\> Grant-SCResource -Resource $Resource -Username "Contoso\Katarina"
```

### 2: Share a resource with a user who is a member of multiple user roles.

The first command gets the template object named Template01 and stores the object in the \$Resource variable.

The second command shares the resource stored in \$Resource (Template01) with the user named Katarina but only while that user is using the ContosoSelfServiceUsers or SelfServiceUserRole02 user roles.

```
PS C:\> $Resource = Get-SCVMTemplate | where {$_.Name -eq "Template01"}
PS C:\> Grant-SCResource -Resource $Resource -Username "Contoso\Katarina" -UserRoleName
@("ContosoSelfServiceUsers", "SelfServiceUserRole02")
```

### 3: Share a resource with all members of a user role.

The first command gets the template object named VMTemplate01 and stores the object in the \$Resource variable.

The second command shares the resource stored in \$Resource (VMTemplate01) with the members of the user role named ContosoSelfServiceUsers.

```
PS C:\> $Resource = Get-SCVMTemplate | where {$_.Name -eq "VMTemplate01"}
PS C:\> Grant-SCResource -Resource $Resource -UserRoleName "ContosoSelfServiceUsers"
```

### 4: Share a resource with all members of a user role by using the user role ID.

The first command gets the template object named VMTemplate01 and stores the object in the \$Resource variable.

---

The second command gets the user role object named ContosoSelfServiceUsers and stores the object in the \$UserRole variable.

The last command shares the resource stored in \$Resource (VMTemplate01) with the members of user role ContosoSelfServiceUsers.

```
PS C:\> $Resource = Get-SCVMTemplate | where {$_.Name -eq "VMTemplate01"}
```

```
PS C:\> $UserRole = Get-SCUserRole -Name "ContosoSelfServiceUsers"
```

```
PS C:\> Grant-SCResource -Resource $Resource -UserRoleId $UserRole.Id
```

## Related topics

[Revoke-SCResource](#)

---

# Import-SCLibraryPhysicalResource

---

## Import-SCLibraryPhysicalResource

Imports a resource into the VMM library.

### Syntax

Parameter Set: Default

```
Import-SCLibraryPhysicalResource [-SourcePath] <String> [[-SharePath] <String> ] [-AllowUnencryptedTransfer] [-OverwriteExistingFiles] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Import-SCLibraryPhysicalResource cmdlet imports a resource into the Virtual Machine Manager (VMM) library. Self-service users can only import resources into their designated user role data path or a folder under that path.

### Parameters

#### -AllowUnencryptedTransfer

Indicates that network file transfers do not require encryption. Allowing unencrypted network file transfers can improve performance if neither the source host nor the destination host requires encryption.

Use this parameter to:

- Allow unencrypted file transfers into, or out of, the library.
- Allow unencrypted file transfers into, out of, or within a host group.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false



---

## **-OverwriteExistingFiles**

Indicates that files with the same name are overwritten when importing or exporting resources into or out of the VMM library.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-SharePath<String>**

Specifies a path to a valid library share on an existing library server that uses a Universal Naming Convention (UNC) path.

Example format: `-SharePath "\\LibServer01\LibShare"`

Aliases	none
Required?	false
Position?	2
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-SourcePath<String>**

Specifies the path to the resource that will be imported.

Aliases	none
Required?	true
Position?	1
Default Value	none

Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Examples**

### **1. Import a library resource by an administrator or delegated administrator.**

This command imports the virtual hard disk named VHD01.vhdd stored in the folder named C:\AdminFolder into the specified library path. If a resource with that name already exists in that library path, this command will overwrite those files.

```
PS C:\> Import-SCLibraryPhysicalResource -SourcePath "C:\AdminFolder\VHD01.vhd" -SharePath
"\\VMMLibraryServer\MSSCVMLibrary\ImportedLibraryResources" -OverwriteExistingFiles
```

### **2. Import a library resource by a self-service user by specifying a path under the default user role data path in the VMM library.**

This command imports the virtual hard disk named VHD01 from the folder named C:\SSFolder to the SSUserSubFolder\Folder01 folder stored under the default user role data path for the self-service user role of which the logged on user is a member.

---

```
PS C:\> Import-SCLibraryPhysicalResource -SourcePath "C:\SSFoldr\VHD01.vhd" -SharePath
"\\<DefaultUserRoleDataPath>\SSUserSubfolder\Folder01"
```

### **3. Import a library resource by a self-service user to the default user role data path in the VMM library.**

This command imports the file named VHD01.vhd stored in C:\SSFoldr to the default user role data path for the self-service user role of which the logged on user is a member.

```
PS C:\> Import-SCLibraryPhysicalResource -SourcePath "C:\SSFoldr\VHD01.vhd"
```

### **Related topics**

[Export-SCLibraryPhysicalResource](#)

---

# Import-SCStorageProvider

---

## Import-SCStorageProvider

Imports storage providers that are installed and registered on a VMM management server.

### Syntax

Parameter Set: Default

```
Import-SCStorageProvider [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously]  
[-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Import-SCStorageProvider cmdlet imports storage providers that are installed and registered on a VMM management server.

Windows Server 2012 offers SMP as a storage provider model for storage array automation. SMP providers are installed locally on the same server running the VMM service. Credentials are not needed to connect to the provider (as with SMI-S providers). After the provider is installed, Import-SCStorageProvider imports those providers into Virtual Machine Manager (VMM).

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Examples

### 1: Import storage providers.

This command imports all storage providers that have been installed and registered on the VMM management server.

```
PS C:\> Import-SCStorageProvider
```

---

# Import-SCTemplate

---

## Import-SCTemplate

Imports a virtual machine template or service template into the VMM library.

### Syntax

Parameter Set: Package

```
Import-SCTemplate -TemplatePackage <Package> [-AllowUnencryptedTransfer] [-Name <String> ]  
[-Overwrite] [-PackageMapping <PackageMapping[]> ] [-Password <String> ] [-Release <String>  
] [-SettingsIncludePrivate] [-SharePath <String> ] [-VMMServer <ServerConnection> ] [  
<CommonParameters>]
```

Parameter Set: Path

```
Import-SCTemplate -Path <String> [-AllowUnencryptedTransfer] [-Name <String> ] [-Overwrite]  
[-PackageMapping <PackageMapping[]> ] [-Password <String> ] [-Release <String> ] [-  
SettingsIncludePrivate] [-SharePath <String> ] [-VMMServer <ServerConnection> ] [  
<CommonParameters>]
```

### Detailed Description

The Import-SCTemplate cmdlet imports a virtual machine template or service template into the VMM library. To export a template out of the library, see Export-SCTemplate.

### Parameters

#### -AllowUnencryptedTransfer

Indicates that network file transfers do not require encryption. Allowing unencrypted network file transfers can improve performance if neither the source host nor the destination host requires encryption.

Use this parameter to:

- Allow unencrypted file transfers into, or out of, the library.
- Allow unencrypted file transfers into, out of, or within a host group.

Aliases	none
Required?	false
Position?	named

Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Overwrite**

Indicates that an import or export operation will overwrite an existing file with the same name. Or, that an import operation will overwrite an existing virtual machine template or service template object with the same name.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-PackageMapping<PackageMapping[]>**

Specifies a package mapping object.

Aliases	none
---------	------



Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Password<String>**

Specifies a secure string that contains a password.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Path<String>**

Specifies the destination path for the operation.

Example formats:

Local path -Path "F:\"

UNC path -Path "\\Library\Templates"

Volume GUID path -Path "\\?\Volume{4703c1ea-8ae7-11db-b473-00123f7603e3}\"

VMware ESX path -Path "[storage1]\MyVMwareFolderForVMs\MyVM.vmx"

Citrix XenServer path - Path "Local storage[99b6212f-b63d-c676-25f9-d6c460992de7]"

Wildcards are supported for "Get" cmdlets and when you specify the UNC path:

Example format:

UNC path -Path "\\VMHostServer\MyVMs\\*VM\*"

Aliases	none
Required?	true
Position?	named
Default Value	none

Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Release<String>**

Specifies a string that describes the release of a library resource. VMM automatically creates a release value for every resource imported into the library. After the resource has been imported, the string can be customized.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-SettingsIncludePrivate**

Indicates that sensitive template settings are included in an import or export operation.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-SharePath<String>**

Specifies a path to a valid library share on an existing library server that uses a Universal Naming Convention (UNC) path.

Example format: `-SharePath "\\LibServer01\LibShare"`

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-TemplatePackage<Package>**

Specifies an exported template package that contains serialized settings of a service or virtual machine template.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

---

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **SCTemplate**

## Examples

### 1: Import a previously exported template package.

The first command gets the exported template package object at the specified path and stores the object in the `$Package` variable.

The second command imports the template package object stored in `$Package`, including all template settings.

```
PS C:\> $Package = Get-SCTemplatePackage -Path  
"C:\TemplateExports\ServiceTemplate01.new.xml"  
PS C:\> Import-SCTemplate -TemplatePackage $Package -SettingsIncludePrivate
```

### 2: Import an export package and specify a new name and release for the imported template.

This command imports the specified template export package with all of the template's settings and specifies a new name and release for the imported template.

```
PS C:\> Import-SCTemplate -Path "C:\TemplateExports\ServiceTemplate01.new.xml" -  
SettingsIncludePrivate -Name "New Service Name" -Release "1.0"
```

### 3. Importing a template that has some/all resources in the exported package while changing mapping

The first command creates a package mapping object for the package stored at the specified path and then stores the package mapping object in the `$Mappings` variable.

The second command gets a mapping object by package ID and stores the object in the `$Mapping` variable.

The third command gets the virtual hard disk object named `VHD01` and stores the object in the `$Resource` variable.

The fourth command binds the mapping stored in `$Mapping` to the object stored in `$Resource` (`VHD01`).

The fifth command sets the package file for the mapping stored in `$Mapping`.

The last command imports the template at the specified path with the specified mappings (in this case, `VHD01` imports to `Share01`).

```
PS C:\> $Mappings = New-SCPackageMapping -Path "C:\TemplateExports\VMTemplate01.xml" -  
PreferPackageResources  
PS C:\> $Mapping = $Mappings | where {$_.PackageID -eq "VHD01.vhd"}  
PS C:\> $Resource = Get-SCVirtualHardDisk -Name "VHD01.vhd"
```

---

```
PS C:\> Set-SCPackageMapping -PackageMapping $Mapping -TargetObject $Resource
PS C:\> Set-SCPackageMapping -PackageMapping $Mapping -PackageFile
"C:\TemplateExports\Resources\VHD01.vhd"
PS C:\> Import-SCTemplate -Path C:\TemplateExports\VMTemplate01.xml -PackageMapping $Mapping
-SharePath "\\LibServer01\Share01"
```

## Related topics

[Export-SCTemplate](#)

[Get-SCTemplatePackage](#)

---

# Install-SCVMHostCluster

---

## Install-SCVMHostCluster

Creates a failover cluster from Hyper-V hosts managed by VMM.

### Syntax

Parameter Set: AddNodes

```
Install-SCVMHostCluster -Credential <VMMCredential> -VMHost <Host[]> -VMHostCluster <HostCluster> [-JobGroup <Guid> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-SkipValidation] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: CreateCluster

```
Install-SCVMHostCluster -ClusterName <String> -Credential <VMMCredential> -VMHost <Host[]> [-ClusterIPAddress <String[]> ] [-ClusterIPAddressPool <StaticIPAddressPool[]> ] [-ClusterReserve <Int32> ] [-Description <String> ] [-JobGroup <Guid> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-SetQuorumDisk <ClientObject> ] [-SetQuorumNodeMajority] [-SkipValidation] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Install-SCVMHostCluster cmdlet creates a failover cluster from Hyper-V hosts managed by Virtual Machine Manager (VMM). Install-SCVMHostCluster can also add a node to an existing cluster.

### Parameters

#### **-ClusterIPAddress<String[]>**

Specifies one or more IP addresses to use as a cluster IP address.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-ClusterIPAddressPool<StaticIPAddressPool[]>**

Specifies a static IP address pool to use as a cluster IP address pool.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-ClusterName<String>**

Specifies the name of a cluster.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-ClusterReserve<Int32>**

Specifies the number of host failures that a host cluster can sustain before VMM designates the cluster as over-committed. The default value is 1.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false

Accept Wildcard Characters?	false
-----------------------------	-------

### **-Credential<VMMCredential>**

Specifies a credential object or, for some cmdlets, a Run As account object that contains the user name and password of an account that has permission to perform this action. Or, in the case of Restart-SCJob, has permission to complete a restarted task.

For more information about the PSCredential object, type: "Get-Help Get-Credential".

For more information about Run As accounts, type: "Get-Help New-SCRunAsAccount".

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Description<String>**

States a description for the specified object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-JobGroup<Guid>**

Specifies an identifier for a series of commands that will run as a set just before the final command that includes the same job group identifier runs.

Aliases	none
---------	------



---

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
---------	------

---

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-SetQuorumDisk<ClientObject>**

Specifies a disk to use as the quorum disk for the cluster.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-SetQuorumNodeMajority**

Sets the quorum mode to Node Majority for the cluster.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-SkipValidation**

Skips cluster validation tests when creating a cluster.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMHost<Host[]>**

Specifies an array of virtual machine host objects.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMHostCluster<HostCluster>**

Specifies a VMM host cluster object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
---------	------

---

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **VMHostCluster**

## Examples

### 1: Create a cluster from managed hosts.

The first command gets the Run As account named RunAsAccount01 and stores it in the \$RunAsAcct variable.

The second command gets the host group object named New York and stores the object in the \$HostGroup variable.

The third command gets the cluster nodes that begin with the name HostClus in the host group stored in \$HostGroup. The command then stores the nodes in the \$Nodes variable.

The last command creates the cluster and names it Cluster01.

```
PS C:\> $RunAsAcct = Get-SCRunAsAccount -Name "RunAsAccount01"
```

```
PS C:\> $HostGroup = Get-SCVMHostGroup -Name "New York"
```

```
PS C:\> $Nodes = Get-SCVMHost | where {$_.Name -like "HostClus*" -and $_.VMHostGroup -eq $HostGroup}
```

```
PS C:\> Install-SCVMHostCluster -VMHost $Nodes -ClusterName "Cluster01" -Credential $RunAsAcct
```

## Related topics

[Add-SCVMHostCluster](#)

[Get-SCVMHostCluster](#)

[Move-SCVMHostCluster](#)

[Read-SCVMHostCluster](#)

---

[Remove-SCVMHostCluster](#)

[Set-SCVMHostCluster](#)

[Test-SCVMHostCluster](#)

[Uninstall-SCVMHostCluster](#)

---

# Invoke-SCPROTip

---

## Invoke-SCPROTip

Performs the action recommended by a PRO tip.

### Syntax

Parameter Set: Default

```
Invoke-SCPROTip [-PROTip] <PROTip> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Invoke-SCPROTip cmdlet performs the action recommended by a Performance and Resource Optimization (PRO) tip. You can use this cmdlet to manually invoke the action recommended by a PRO tip that is not set to be implemented automatically.

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

#### **-PROTip<PROTip>**

Specifies a PRO tip object.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

---

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- PROTip

## Examples

### 1: Invoke the first active PRO tip.

The first command gets all active PRO tip objects from the VMM database and stores the objects in the \$PROTips object array.

The second command implements the suggested action for the first PRO tip stored in \$PROTips (designated by the [0]).

```
PS C:\> $PROTips = Get-SCPROTip
```

```
PS C:\> Invoke-SCPROTip -PROTip $PROTips[0]
```

## Related topics

[Clear-SCPROTip](#)

[Get-SCPROTip](#)

[Set-SCPROTip](#)

[Test-SCPROTip](#)



---

# Invoke-SCScriptCommand

---

## Invoke-SCScriptCommand

Runs a script command on the specified host.

### Syntax

Parameter Set: VMHost

```
Invoke-SCScriptCommand -Executable <String> -VMHost <Host> [-CommandParameters <String> ] [-LibraryResource <CustomResource> ] [-RunAsAccount <RunAsAccount> ] [-ScriptCommandSetting <SCScriptCommandSetting> ] [-StandardInput <String> ] [-TimeoutSeconds <Int32> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Invoke-SCScriptCommand cmdlet runs a script command on the specified host. This cmdlet is only supported on Hyper-V hosts.

### Parameters

#### **-CommandParameters<String>**

Specifies the parameters for a script or executable program.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

#### **-Executable<String>**

Specifies the name of an executable program.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-LibraryResource<CustomResource>**

Specifies a resource stored in the VMM library.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-RunAsAccount<RunAsAccount>**

Specifies a Run As account that contains credentials with permission to perform this action.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-ScriptCommandSetting<SCScriptCommandSetting>**

Specifies a script command setting object.

---

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-StandardInput<String>**

Specifies a path to a file that contains standard input information to use with the script command.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-TimeoutSeconds<Int32>**

Specifies the amount of time, in seconds, that a process waits before timing out.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMHost<Host>**

Specifies a virtual machine host object. VMM supports Hyper-V hosts, VMware ESX hosts, and Citrix XenServer hosts.

---

For more information about each type of host, type: "Get-Help Add-SCVMHost -detailed". See the examples for a specific cmdlet to determine how that cmdlet uses this parameter.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

### **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **ScriptCommand**

### **Notes**

- Requires a VMHost object, which can be retrieved by using the Get-SCVMHost cmdlet.

---

## Examples

### 1. Run a script command.

The first command gets the host object named VMHost01 and stores the object in the \$VMHost variable.

The second command runs the executable program named cmd.exe with the specified parameters on the host stored in \$VMHost. In this case, Invoke-SCScriptCommand removes the test directory from the c: drive on VMHost01.

```
PS C:\> $VMHost = Get-SCVMHost -ComputerName "VMHost01"
```

```
PS C:\> Invoke-SCScriptCommand -VMHost $VMHost -Executable "cmd.exe" -CommandParameters "/C rd C:\test" -TimeoutSeconds 60
```

### Related topics

[Add-SCScriptCommand](#)

[Get-SCScriptCommand](#)

[Get-SCScriptCommandSetting](#)

[Get-SCVMHost](#)

[New-SCScriptCommandSetting](#)

[Remove-SCScriptCommand](#)

[Set-SCScriptCommand](#)

---

# Mount-SCStorageDisk

---

## Mount-SCStorageDisk

Mounts a storage disk.

### Syntax

Parameter Set: Default

```
Mount-SCStorageDisk -StorageDisk <StorageDisk> [-CreateClusterSharedVolume] [-DeleteClusterSharedVolume] [-JobGroup <Guid> ] [-JobVariable <String> ] [-MountPoint <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-StorageLogicalUnit <StorageLogicalUnit> ] [ <CommonParameters>]
```

Parameter Set: ClusterFormatGPT

```
Mount-SCStorageDisk -GuidPartitionTable -JobGroup <Guid> -StorageLogicalUnit <StorageLogicalUnit> [-CreateClusterSharedVolume] [-DeleteClusterSharedVolume] [-DesiredUnitAllocationSizeBytes <UInt32> ] [-ForceFormat] [-FullFormat] [-JobVariable <String> ] [-MountPoint <String> ] [-PROTipID <Guid> ] [-QuickFormat] [-RunAsynchronously] [-VolumeLabel <String> ] [ <CommonParameters>]
```

Parameter Set: ClusterFormatMBR

```
Mount-SCStorageDisk -JobGroup <Guid> -MasterBootRecord -StorageLogicalUnit <StorageLogicalUnit> [-CreateClusterSharedVolume] [-DeleteClusterSharedVolume] [-DesiredUnitAllocationSizeBytes <UInt32> ] [-ForceFormat] [-FullFormat] [-JobVariable <String> ] [-MountPoint <String> ] [-PROTipID <Guid> ] [-QuickFormat] [-RunAsynchronously] [-VolumeLabel <String> ] [ <CommonParameters>]
```

Parameter Set: ForceNewDiskIdGuid

```
Mount-SCStorageDisk -DiskId <Guid> -StorageDisk <StorageDisk> [-CreateClusterSharedVolume] [-DeleteClusterSharedVolume] [-JobGroup <Guid> ] [-JobVariable <String> ] [-MountPoint <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

Parameter Set: ForceNewDiskIdSignature

```
Mount-SCStorageDisk -DiskSignature <String> -StorageDisk <StorageDisk> [-CreateClusterSharedVolume] [-DeleteClusterSharedVolume] [-JobGroup <Guid> ] [-JobVariable <String> ] [-MountPoint <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

Parameter Set: FullFormatGPT

```
Mount-SCStorageDisk -FullFormat -GuidPartitionTable -StorageDisk <StorageDisk> [-CreateClusterSharedVolume] [-DeleteClusterSharedVolume] [-DesiredUnitAllocationSizeBytes <UInt32> ] [-ForceFormat] [-JobGroup <Guid> ] [-JobVariable <String> ] [-MountPoint <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VolumeLabel <String> ] [ <CommonParameters>]
```

Parameter Set: FullFormatMBR

```
Mount-SCStorageDisk -FullFormat -MasterBootRecord -StorageDisk <StorageDisk> [-CreateClusterSharedVolume] [-DeleteClusterSharedVolume] [-DesiredUnitAllocationSizeBytes
```

---

```
<UInt32> ] [-ForceFormat] [-JobGroup <Guid> ] [-JobVariable <String> ] [-MountPoint <String>
] [-PROTipID <Guid> ] [-RunAsynchronously] [-VolumeLabel <String> ] [ <CommonParameters>]
```

Parameter Set: QuickFormatGPT

```
Mount-SCStorageDisk -GuidPartitionTable -QuickFormat -StorageDisk <StorageDisk> [-
CreateClusterSharedVolume] [-DeleteClusterSharedVolume] [-DesiredUnitAllocationSizeBytes
<UInt32> ] [-ForceFormat] [-JobGroup <Guid> ] [-JobVariable <String> ] [-MountPoint <String>
] [-PROTipID <Guid> ] [-RunAsynchronously] [-VolumeLabel <String> ] [ <CommonParameters>]
```

Parameter Set: QuickFormatMBR

```
Mount-SCStorageDisk -MasterBootRecord -QuickFormat -StorageDisk <StorageDisk> [-
CreateClusterSharedVolume] [-DeleteClusterSharedVolume] [-DesiredUnitAllocationSizeBytes
<UInt32> ] [-ForceFormat] [-JobGroup <Guid> ] [-JobVariable <String> ] [-MountPoint <String>
] [-PROTipID <Guid> ] [-RunAsynchronously] [-VolumeLabel <String> ] [ <CommonParameters>]
```

## Detailed Description

The Mount-SCStorageDisk cmdlet mounts a storage disk.

## Parameters

### -CreateClusterSharedVolume

Indicates that the cmdlet will create a Cluster Shared Volume.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### -DeleteClusterSharedVolume

Indicates that the cmdlet will delete a Cluster Shared Volume.

Aliases	none
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-DesiredUnitAllocationSizeBytes<UInt32>**

Specifies, in bytes, the default allocation size of a volume.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-DiskId<Guid>**

Specifies the ID of a disk object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-DiskSignature<String>**

Specifies the signature of a disk object.

Aliases	none
Required?	true



---

Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-ForceFormat**

Forces the formatting of the storage disk even if volumes are already present.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-FullFormat**

Indicates that a full format of the partition is performed.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-GuidPartitionTable**

Indicates that the storage disk is a GUID partition table disk.

Aliases	none
Required?	true

---

Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-JobGroup<Guid>**

Specifies an identifier for a series of commands that will run as a set just before the final command that includes the same job group identifier runs.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-MasterBootRecord**

Indicates that the storage disk is a master boot record disk.

Aliases	none
---------	------

---

Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-MountPoint<String>**

Specifies a mount point location.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-QuickFormat**

Indicates that a quick format of the partition is performed.

Aliases	none
---------	------

Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-StorageDisk<StorageDisk>**

Specifies a disk on a Hyper-V or VMware ESX host that a virtual machine on that host can use instead of using a virtual hard disk. This disk is referred to as a pass-through disk (the corresponding VMware term is Raw Device Mapping, or RDM). The host disk is either a local hard disk or a logical unit on a Storage Area Network (SAN). VMM lets the virtual machine bypass the host's file system and access the pass-through disk directly.

TYPE OF HOST    PASS-THROUGH DISK SUPPORT

-----

Hyper-V        Supports pass-through disks

Supports converting a pass-through disk to a VHD

VMware ESX    Supports pass-through disks (RDP), but not disk conversion

Citrix XenServer Does not support pass-through disks

Aliases	none
Required?	true
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-StorageLogicalUnit<StorageLogicalUnit>**

Specifies a storage logical unit object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-VolumeLabel<String>**

Specifies a label for a disk volume.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

### **Notes**

- Requires a VMM storage disk object, which can be retrieved using the Get-SCStorageDisk cmdlet.

---

## Examples

### 1: Format a new disk.

The first command generates a globally unique identifier (GUID) and stores the GUID string in variable `$JobGroup`. Subsequent commands that include this GUID are collected into a single job group.

The second command gets the host object named `VMHost01` and stores the object in the `$VMHost` variable.

The third command gets the storage logical unit object named `LUN01` and stores the object in the `$LU` variable.

The fourth command registers `LUN01` with `VMHost01`. Using the `JobGroup` parameter specifies that this command will not run until just before the final command that includes the `JobGroup` with the same GUID.

The fifth command mounts `LUN01` on `VMHost01`, performs a quick format on the volume, labels the volume "New Volume", and sets the mount point to `S:\`. Using the `JobGroup` parameter specifies that this command will not run until just before the final command that includes the `JobGroup` with the same GUID.

The last command updates `VMHost01` with the mounted storage disk. This command uses the `JobGroup` parameter to register and mount `LUN01` prior to running `Set-SCVMHost`.

```
PS C:\> $JobGroup = [Guid]::NewGuid().ToString()
PS C:\> $VMHost = Get-SCVMHost -ComputerName "VMHost01"
PS C:\> $LU = Get-SCStorageLogicalUnit -Name "LUN01"
PS C:\> Register-SCStorageLogicalUnit -StorageLogicalUnit $LU -VMHost $VMHost -JobGroup
$JobGroup
PS C:\> Mount-SCStorageDisk -QuickFormat -MasterBootRecord -VolumeLabel "New Volume" -
StorageLogicalUnit $LU -MountPoint "S:\" -JobGroup $JobGroup
PS C:\> Set-SCVMHost -VMHost $VMHost -JobGroup $JobGroup
```

---

# Move-SCVirtualHardDisk

---

## Move-SCVirtualHardDisk

Moves a virtual hard disk file from one location to another on the same host or, when used with Move-SCVirtualMachine, to a location on a different host.

### Syntax

Parameter Set: MoveVHDonIDE

```
Move-SCVirtualHardDisk -Bus <Byte> -IDE -JobGroup <Guid> -LUN <Byte> -Path <String> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: MoveVHDonSCSI

```
Move-SCVirtualHardDisk -Bus <Byte> -JobGroup <Guid> -LUN <Byte> -Path <String> -SCSI [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: SetVM

```
Move-SCVirtualHardDisk [-VirtualHardDisk] <StandaloneVirtualHardDisk> -Path <String> [-JobGroup <Guid> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

### Detailed Description

The Move-SCVirtualHardDisk cmdlet moves a Windows-based virtual hard disk file (a .vhd file) or a VMware-based virtual hard disk file (a .vmdk file) from one location to another on the same host. You can also use Move-SCVirtualHardDisk with the Move-SCVirtualMachine cmdlet to move a virtual hard disk file to a location on a different host.

You can use this cmdlet to perform the following tasks:

- Move a virtual hard disk on a running VMware virtual machine with no service interruption.
- Move a virtual hard disk on a running Windows Server 2008 R2 or Windows Server 2012 virtual machine with minimal service interruption.

In this case, use Move-SCVirtualHardDisk with Move-SCVirtualMachine.

- Move a virtual hard disk on a virtual machine on any type of host if the virtual machine is in a stopped state or in a saved state. In this case, use Move-SCVirtualHardDisk with Set-SCVirtualMachine. This option is supported if either of the

---

following conditions are true:

- The virtual machine is on a host that uses Windows Server 2008 R2 or Windows Server 2012 Hyper-V technology or on a VMware ESX host, and the virtual machine is in a saved state or in a stopped state.
- The virtual machine is on any other supported host, and it is in a stopped state.

Usage examples:

- If a host has multiple physical disk drives and the virtual machine has two virtual hard disks (one disk might contain the operating system and the other disk might contain data), you can use this cmdlet to move one of the virtual hard disks to a different physical hard drive in order to improve performance for both virtual hard disks.
- If the virtual machine has one dynamically expanding virtual hard disk and you discover that the virtual hard disk has expanded to a point where it uses most of the space on its current physical hard disk on the host, you can use this cmdlet to move the expanded virtual hard disk to a larger physical hard disk if one is available on the host.

## Parameters

### **-Bus<Byte>**

Specifies the IDE bus to which to attach a virtual disk drive or virtual DVD drive, or the SCSI bus to which to attach a virtual disk drive.

Example format: -IDE -Bus 1 -LUN 0

Example format: -SCSI -Bus 0 -LUN 1

NOTE: See the examples for a specific cmdlet to determine how that cmdlet uses this parameter.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false



---

## **-IDE**

Specifies IDE as the bus type to which to attach a virtual disk drive object or a virtual DVD drive object configured on a virtual machine or on a template.

Example format: -IDE –Bus 0 –LUN 1

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-JobGroup<Guid>**

Specifies an identifier for a series of commands that will run as a set just before the final command that includes the same job group identifier runs.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false

Accept Wildcard Characters?	false
-----------------------------	-------

### **-LUN<Byte>**

Specifies the logical unit number (LUN) for a virtual disk drive object or for a virtual DVD drive object on an IDE bus, or for a virtual disk drive object on a SCSI bus.

Example format: -IDE -Bus 1 -LUN 0

Example format: -SCSI -Bus 0 -LUN 1

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Path<String>**

Specifies the destination path for the operation.

Example formats:

Local path -Path "F:\"

UNC path -Path "\\Library\Templates"

Volume GUID path -Path "\\?\Volume{4703c1ea-8ae7-11db-b473-00123f7603e3}\"

VMware ESX path -Path "[storage1]MyVMwareFolderForVMs\MyVM.vmx"

Citrix XenServer path - Path "Local storage[99b6212f-b63d-c676-25f9-d6c460992de7]"

Wildcards are supported for "Get" cmdlets and when you specify the UNC path:

Example format:

UNC path -Path "\\VMHostServer\MyVMs\\*VM\*"

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-SCSI**

Specifies SCSI as the bus type to which to attach a virtual disk drive object configured on a virtual machine or on a template.

Example format: -SCSI -Bus 0 -LUN 0

Aliases	none
Required?	true
Position?	named
Default Value	none

Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VirtualHardDisk<StandaloneVirtualHardDisk>**

Specifies a virtual hard disk object.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **VirtualHardDisk**

---

## Examples

### 1: Move a virtual hard disk file from one location to another on the same host.

The first command gets the virtual machine object named VM01 and stores the object in the \$VM variable.

The second command stores the first virtual hard disk object on VM01 in the \$VHD variable.

The last command moves the virtual hard disk stored in \$VHD to the existing folder "C:\VHDs".

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM01"
```

```
PS C:\> $VHD = $VM.VirtualHardDisks[0]
```

```
PS C:\> Move-SCVirtualHardDisk -VirtualHardDisk $VHD -Path "C:\VHDs"
```

## Related topics

[Compress-SCVirtualDiskDrive](#)

[Copy-SCVirtualHardDisk](#)

[Get-SCVirtualHardDisk](#)

[Remove-SCVirtualHardDisk](#)

[Set-SCVirtualHardDisk](#)

---

# Move-SCVirtualMachine

---

## Move-SCVirtualMachine

Moves a virtual machine stored in the VMM library or deployed on a host to a new location on a host.

### Syntax

Parameter Set: Default

```
Move-SCVirtualMachine [-VM] <VM> [-BlockLiveMigrationIfHostBusy] [-HighlyAvailable <Boolean> ] [-JobGroup <Guid> ] [-JobVariable <String> ] [-Path <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-StartVMOnTarget] [-UseCluster] [-UseLAN] [-VMHost <Host> ] [ <CommonParameters>]
```

### Detailed Description

The Move-SCVirtualMachine cmdlet moves a virtual machine stored in the Virtual Machine Manager (VMM) library or deployed on a host to a new location on a host.

In System Center 2012 Service Pack 1 (SP1), you can take advantage of new migration capabilities included in Windows Server 2012, such as live migration of virtual machines between two standalone computers, or live migration between standalone computers and a cluster node. Additionally, multiple concurrent live migrations are supported. For more information about migrating virtual machines in System Center 2012 SP1, see [Migrating virtual machines and storage in the TechNet Library](#).

**NOTE:** If you move a virtual machine deployed on a host running Windows Server 2008 R2 to a host running Windows Server 2012, you cannot move the virtual machine back to a host running Windows Server 2008 R2.

VMM includes storage migration features that let you move one or more virtual hard disks of a running virtual machine to a new location. You can use the Move-SCVirtualMachine cmdlet with the Move-SCVirtualHardDisk cmdlet to move Windows-based virtual hard disk (.vhd) files and VMware-based virtual hard disk (.vmdk) files to a location on a different host. If the virtual machine is deployed on a host running Windows Server 2008 R2 or Windows Server 2012 the virtual machine experiences minimal service interruption. If the virtual machine is deployed on a VMware ESX host, VMM uses VMware VMotion so that no service interruption occurs. You can also use the Move-SCVirtualHardDisk cmdlet to move a .vhd file or a .vmdk file from one location to another on the same host.

**NOTE:** To move a virtual machine from a host and store it in the library, you must use the Save-SCVirtualMachine cmdlet.

VMM can use any of the following transfer methods (listed in the order in which VMM tries to use them):

- Hyper-V LIVE MIGRATION – If a virtual machine is running and is deployed on a Hyper-V host that is a node of a Windows Server 2008 R2 or Windows Server 2012 host cluster, by default, VMM will use Hyper-V live

---

migration to move the virtual machine to another node in the cluster without any disruption of service, for example, moving a running virtual machine will not disconnect it from the network. Additionally, the virtual machine will retain its high availability attribute on the virtual machine. You do not need to specify a path for this type of move. You can start live migration of multiple virtual machines at the same time.

- WINDOWS SERVER 2008 CLUSTER MIGRATION – System Center 2012 continues to support Windows 2008 Cluster

Migration (sometimes called Quick Migration). Cluster Migration lets you move a running virtual machine on a Hyper-V node of a host cluster. It also lets you move a virtual machine that is in a stopped or saved state and that is deployed to another node in the cluster. You can use Cluster Migration to move a virtual machine in a stopped or saved state if the virtual machine is deployed on either of the following nodes:

- \* A node in a Windows Server 2008 cluster

- \* A node in a Windows Server 2008 R2 cluster

You do not need to specify a path for this type of move. Windows Server 2008 Cluster Migration places the virtual machine in a saved state during migration, which causes a temporary loss of service to any users of that virtual machine.

- VMWARE LIVE MIGRATION – If a virtual machine deployed on a VMware ESX host uses shared storage, VMM can use the VMware live migration feature (VMware VMotion) to move the virtual machine to a new host.

You do not need to specify a path for this type of move. The Move-SCVirtualMachine cmdlet can use VMware VMotion to move a virtual machine from one ESX host to another only if both ESX hosts are in the same Datacenter container on the vCenter Server.

- CITRIX XENSERVER XENMOTION – If a virtual machine deployed on a Citrix XenServer host uses shared storage and is part of the same Resource Pool, VMM can use the XenServer live migration feature (Citrix XenMotion) to move the virtual machine to a new host.

You do not need to specify a path for this type of move. The Move-SCVirtualMachine cmdlet can use Citrix XenMotion to move a virtual machine from one XenServer host to another only if both XenServer hosts are in the same Resource Pool.

This is the only supported method for moving a virtual machine

---

directly between XenServer hosts in VMM.

- SAN MIGRATION (Fibre Channel, iSCSI, or NPIV) – If the virtual machine is on a host that is connected to a SAN and the virtual machine is on a SAN LUN, VMM can move that virtual machine to another host if that host has access to the same SAN. In a SAN transfer, the target LUNs are redirected from the source host to the destination host (no files are moved), which is why a SAN transfer is much faster than moving virtual machine files from one host to another over a local area network (LAN). VMM supports SAN migration of virtual machines into and out of a cluster.

You must specify a path for this type of move. VMM can use an NPIV SAN transfer if a host bus adapter (HBA) with NPIV support is available.

- NETWORK MIGRATION – If no faster method is available, VMM uses a network transfer to move the virtual machine files from one host to another over the LAN that connects the two hosts. You can choose to use this transfer type even if the SAN transfer type is available. You must specify a path for this type of move.

When more than one transfer type is available, the Move-SCVirtualMachine cmdlet automatically uses the fastest available transfer type to move a virtual machine. If the first method is not appropriate or available for the virtual machine you want to move, VMM tries to use the next method, and so on. If you want to force the use of a network transfer, specify the UseLAN parameter.

## Parameters

### **-BlockLiveMigrationIfHostBusy**

Blocks retrying a Hyper-V live migration if the migration failed because the source host or the destination host is already participating in another live migration.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false



---

## **-HighlyAvailable<Boolean>**

Specifies that a virtual machine will be placed on a Hyper-V host that is part of a host cluster. Configure this setting on a virtual machine, or on a template or hardware profile that will be used to create virtual machines.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-JobGroup<Guid>**

Specifies an identifier for a series of commands that will run as a set just before the final command that includes the same job group identifier runs.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-Path<String>**

Specifies the destination path for the operation.

Example formats:

Local path -Path "F:\"

UNC path -Path "\\Library\Templates"

Volume GUID path -Path "\\?\Volume{4703c1ea-8ae7-11db-b473-00123f7603e3}\"

VMware ESX path -Path "[storage1]\MyVMwareFolderForVMs\MyVM.vmx"

Citrix XenServer path - Path "Local storage[99b6212f-b63d-c676-25f9-d6c460992de7]"

Wildcards are supported for "Get" cmdlets and when you specify the UNC path:

Example format:

UNC path -Path "\\VMHostServer\MyVMs\\*VM\*"

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

---

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-StartVMOnTarget**

Specifies that a virtual machine starts as soon as it reaches its destination host.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-UseCluster**

Forces the use of Cluster Migration for the transfer of a virtual machine that is in a saved state to a host, even if the cluster supports Hyper-V live migration.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-UseLAN**

Forces a transfer over the local area network (LAN) even if a faster transfer mechanism, such as a storage area network (SAN) transfer, is available.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VM<VM>**

Specifies a virtual machine object.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **-VMHost<Host>**

Specifies a virtual machine host object. VMM supports Hyper-V hosts, VMware ESX hosts, and Citrix XenServer hosts.

For more information about each type of host, type: "Get-Help Add-SCVMHost -detailed". See the examples for a specific cmdlet to determine how that cmdlet uses this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)

---

Accept Wildcard Characters?
-----------------------------

false
-------

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **VirtualMachine**

## Examples

### 1: Move a virtual machine from the library to a host.

The first command gets the virtual machine object named VM01, which is currently stored in the VMM library on the library server named LibServer01, and stores the object in the \$VM variable. This example assumes that only one virtual machine named VM01 is currently stored on LibServer01, and that, therefore, Get-SCVirtualMachine retrieves only one object.

The second command gets the host object named VMHost01 and stores the object in the \$VMHost variable.

The last command moves the virtual machine from its current location in the library to the location D:\VirtualMachinePath on the host stored in \$VMHost. The command automatically uses the fastest available transfer type. When the command completes, it returns information about the moved virtual machine.

```
PS C:\> $VM = Get-SCVirtualMachine | where { $_.Name -eq "VM01" -and $_.LibraryServer -eq "LibServer01.Contoso.com" }
PS C:\> $VMHost = Get-SCVMHost -ComputerName "VMHost01.Contoso.com"
PS C:\> Move-SCVirtualMachine -VMHost $VMHost -VM $VM -Path "D:\VirtualMachinePath"
```

### 2: Move a virtual machine from the library to a host asynchronously.

The first two commands in this example are identical to the commands in example 1, except for the name of the VM host.

When the third command moves the virtual machine from its current location to D:\VirtualMachinePath on VMHost02, it uses the RunAsynchronously parameter to return control to the command shell immediately, and the JobVariable parameter to track the progress of the job. JobVariable stores a record of the job progress in the MoveVMJob variable. For the JobVariable parameter, you do not use the dollar sign (\$) when the variable is created.

The last command displays the contents of \$MoveVMJob, which includes a description of the move job, its status, its progress, and other information.

---

```
PS C:\> $VM = Get-SCVirtualMachine | where { $_.Name -eq "VM01" -and $_.LibraryServer -eq "LibServer01.Contoso.com" }
PS C:\> $VMHost = Get-SCVMHost -ComputerName "VMHost02.Contoso.com"
PS C:\> Move-SCVirtualMachine -VMHost $VMHost -VM $VM -Path "D:\VirtualMachinePath" -RunAsynchronously -JobVariable "MoveVMJob"
PS C:\> $MoveVMJob
```

### **3: Move a virtual machine from the library to a host by forcing a LAN transfer.**

The first command gets the virtual machine object named VM03 on library server LibServer01 and stores the object in the \$VM variable.

The second command gets the host object named VMHost03 and stores the object in the \$VMHost variable.

The last command moves the virtual machine VM03 from its current location in the library to D:\VirtualMachinePath on VMHost03 using the UseLAN parameter to specify that the transfer use a network transfer even if faster transfer mechanisms are available.

```
PS C:\> $VM = Get-SCVirtualMachine | where { $_.Name -eq "VM03" -and $_.LibraryServer -eq "LibServer01.Contoso.com" }
PS C:\> $VMHost = Get-SCVMHost -ComputerName "VMHost03.Contoso.com"
PS C:\> Move-SCVirtualMachine -VMHost $VMHost -VM $VM -Path "D:\VirtualMachinePath" -UseLAN
```

### **4: Move a virtual machine between hosts by using VMware VMotion.**

The first gets the virtual machine object named VM04 on ESXHost01 and stores the object in the \$VM variable.

The second command gets the ESX host object named ESXHost02 and stores the object in the \$VMHost variable.

In the last command, the Move-SCVirtualMachine cmdlet uses VMware VMotion to move the virtual machine from its current ESX host to the other ESX host.

NOTE: The Move-SCVirtualMachine cmdlet can use the VMware VMotion feature to move a virtual machine from one ESX host to another only if both ESX servers are in the same Datacenter container on the vCenter Server.

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM04" | where { $_.VMHost.Name -eq "ESXHost01" }
PS C:\> $VMHost = Get-SCVMHost | where { $_.Name -eq "ESXHost02" }
PS C:\> Move-SCVirtualMachine -VM $VM -VMHost $VMHost -Path "[Storage2]"
```

---

## 5: Move a highly available virtual machine between nodes in a host cluster by using Hyper-V live migration.

The first command gets the virtual machine object named HAVM05 on VMHVHostNode05A and stores the in the \$VM variable. This example assumes that HAVM05 is a highly available virtual machine and that VMHVHostNode05A and VMHVHostNode05B are nodes in a Hyper-V host cluster.

The second command gets the host object named VMHVHostNode05B and stores the object in the \$VMHost variable.

In the last command, the Move-SCVirtualMachine cmdlet uses live migration to move the virtual machine from VMHVHostNode05A to VMHVHostNode05B.

```
PS C:\> $VM = Get-SCVirtualMachine -Name "HAVM05" | where {$_.VMHost.Name -eq "VMHVHostNode05A.Contoso.com"}
PS C:\> $VMHost = Get-SCVMHost | where {$_.Name -eq "VMHVHostNode05B.Contoso.com"}
PS C:\> Move-SCVirtualMachine -VM $VM -VMHost $VMHost -Path "D:\VMs\"
```

## 6: Move a running virtual machine on a Hyper-V host to a new location on the same host.

The first command stores the string "E:\VHDs" in \$MoveVhdPath. This is the path to which you want to move the virtual hard disk of the virtual machine.

The second command gets the virtual machine object named VM06 and stores the object the \$VM variable.

The third command gets the host object named VMHost06 and stores the object in the \$VMHost variable. This example assumes that VMHost06 is a Hyper-V host.

The fourth command stores the string "E:\VirtualMachinePath" in the \$HostPath variable. This is the path to which you want to move VM06.

The fifth command creates a new GUID string and stores it in the \$JobGroupID variable. This GUID is a job group ID that functions as an identifier that groups subsequent commands that include this identifier into a single job group.

The sixth command sets the virtual hard disk path to the value stored in \$MoveVhdPath and it connects the virtual hard disk to Bus 1 and LUN 1 on the virtual IDE controller on the virtual machine. This command uses the JobGroup parameter to specify that it does not actually run until the Move-SCVirtualMachine cmdlet triggers the running of any commands in the JobGroup list.

The last command runs any commands that contain \$JobGroupID (in this case, Move-SCVirtualHardDisk), and it runs Move-SCVirtualMachine. As a result, VM06 (a running virtual machine) is moved from its current location to E:\VirtualMachinePath on the same host. The virtual machine's virtual hard disk is moved to E:\VHDs.

NOTE: You can also move a running virtual machine from one Hyper-V host to another Hyper-V host.

```
PS C:\> $MoveVhdPath = "E:\VHDs"
PS C:\> $VM = Get-SCVirtualMachine "VM06"
PS C:\> $VMHost = Get-SCVMHost "VMHost06"
PS C:\> $HostPath = "E:\VirtualMachinePath"
PS C:\> $JobGroupID = [System.Guid]::NewGuid().ToString()
```

---

```
PS C:\> Move-SCVirtualHardDisk -IDE -Bus 1 -Lun 1 -Path $MoveVhdPath -JobGroup $JobGroupID
PS C:\> Move-SCVirtualMachine -VM $VM -VMHost $VMHost -Path $HostPath -JobGroup $JobGroupID
```

## Related topics

[Get-SCVirtualMachine](#)

[Read-SCVirtualMachine](#)

[Save-SCVirtualMachine](#)



---

# Move-SCVMHost

---

## Move-SCVMHost

Moves a virtual machine host managed by VMM from one host group to another.

### Syntax

Parameter Set: Default

```
Move-SCVMHost [-VMHost] <Host> -ParentHostGroup <HostGroup> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

### Detailed Description

The Move-SCVMHost cmdlet moves one or more virtual machine hosts managed by Virtual Machine Manager (VMM) from its current host group to a new parent host group. Before you can move a host, its new parent host group must exist.

If the host is a computer that is managed by members of a Self-Service User or Delegated Administrator user role, moving the host from one host group to another might affect the roles that have access to the host or to virtual machines on that host.

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-ParentHostGroup<HostGroup>**

Specifies the parent host group that contains one or more hosts, host groups, or host clusters.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-VMHost<Host>**

Specifies a virtual machine host object. VMM supports Hyper-V hosts, VMware ESX hosts, and Citrix XenServer hosts.

For more information about each type of host, type: "Get-Help Add-SCVMHost -detailed". See the examples for a specific cmdlet to determine how that cmdlet uses this parameter.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **Host**

## **Notes**

- Requires a VMM host object, which can be retrieved by using the Get-SCVMHost cmdlet.

## **Examples**

### **1: Move a single host to a new parent host group.**

The first command gets the host named VMHost01 and stores it in the \$VMHost variable.

The second command gets the host group object named NewHostGroup and stores it in the \$NewHG variable.

The last command moves the host stored in the \$VMHost variable (VMHost01) to the host group stored in the \$NewHG variable (NewHostGroup).

```
PS C:\> $VMHost = Get-SCVMHost -ComputerName "VMHost01"
```

```
PS C:\> $NewHG = Get-SCVMHostGroup -Name "NewHostGroup01"
```

```
PS C:\> Move-SCVMHost -VMHost $VMHost -ParentHostGroup $NewHG
```

---

## 2: Move all hosts to a new parent host group.

The first command gets all host objects and stores them in the `$AllHosts` array.

The second command gets the host group named "NewHostGroup02" and stores it in the `$NewHG` variable.

The last command moves each host object in the `$AllHosts` array to the new parent host group stored in the `$NewHG` variable (NewHostGroup02).

```
PS C:\> $AllHosts = Get-SCVMHost
PS C:\> $NewHG = Get-SCVMHostGroup -Name "NewHostGroup02"
PS C:\> $AllHosts | Move-SCVMHost -ParentHostGroup $NewHG
```

## 3: Move a set of hosts from one host group to a new parent host group.

The first command gets all host objects whose host group contains the string "OldGroup" and stores them in the `$SpecificHosts` variable.

The second command gets the host group named "NewHostGroup03" and stores it in the `$NewHG` variable.

The last command moves each host object stored in `$SpecificHosts` to the host group stored in the `$NewHG` variable (NewHostGroup03).

```
PS C:\> $SpecificHosts = Get-SCVMHost | where { $_.VMHostGroup -like "*OldGroup*" }
PS C:\> $NewHG = Get-SCVMHostGroup -Name "NewHostGroup03"
PS C:\> $SpecificHosts | Move-SCVMHost -ParentHostGroup $NewHG
```

## Related topics

[Add-SCVMHost](#)

[Get-SCVMHost](#)

[Move-SCVMHostGroup](#)

[New-SCVMHostGroup](#)

[Read-SCVMHost](#)

[Remove-SCVMHost](#)

[Set-SCVMHost](#)

---

# Move-SCVMHostCluster

---

## Move-SCVMHostCluster

Moves a host cluster object managed by VMM from one host group to another.

### Syntax

Parameter Set: Default

```
Move-SCVMHostCluster [-VMHostCluster] <HostCluster> -ParentHostGroup <HostGroup> [-  
JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

### Detailed Description

The Move-SCVMHostCluster cmdlet moves a host cluster object managed by Virtual Machine Manager (VMM) from one host group to another.

You cannot use the Move-SCVMHostCluster cmdlet to move a VMware host cluster. Instead, use vCenter Server to move a VMware host cluster.

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

#### **-ParentHostGroup<HostGroup>**

Specifies the parent host group that contains one or more hosts, host groups, or host clusters.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMHostCluster<HostCluster>**

Specifies a VMM host cluster object.

---

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **VMHostCluster**

## Examples

### 1: Move a specified host cluster to a new parent host group.

The first command gets the host cluster object named VMHostCluster01.Contoso.com and stores the object in the \$VMHostCluster variable.

The second command gets the host group object named Production and stores the object in the \$DestinationHG variable.

The last command moves host cluster VMHostCluster01.Contoso.com, stored in \$VMHostCluster, from its current host group to the host group named Production, stored in \$DestinationHG.

```
PS C:\> $VMHostCluster = Get-SCVMHostCluster -Name "VMHostCluster01.Contoso.com"
```

```
PS C:\> $DestinationHG = Get-SCVMHostGroup -Name "Production"
```

```
PS C:\> Move-SCVMHostCluster -VMHostCluster $VMHostCluster -ParentHostGroup $DestinationHG
```

## Related topics

[Add-SCVMHostCluster](#)

[Find-SCCluster](#)

[Get-SCVMHostCluster](#)

[Install-SCVMHostCluster](#)

[Read-SCVMHostCluster](#)

[Remove-SCVMHostCluster](#)

---

[Set-SCVMHostCluster](#)

[Test-SCVMHostCluster](#)

[Uninstall-SCVMHostCluster](#)



---

# Move-SCVMHostGroup

---

## Move-SCVMHostGroup

Moves a host group from the current location to a new location under a different host group parent.

### Syntax

Parameter Set: Default

```
Move-SCVMHostGroup [-VMHostGroup] <HostGroup> -ParentHostGroup <HostGroup> [-JobGroup <Guid> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

### Detailed Description

The Move-SCVMHostGroup cmdlet moves one or more host group objects, which contain virtual machine hosts managed by Virtual Machine Manager (VMM), from the current location to a new location under a different host group parent. You can place host groups under the default root host group (All Hosts) or under any other host group created by an administrator.

All hosts within a moved host group acquire a new host path relative to the root host group. Changing the structure of host groups might change which Self Service User or Delegated Administrator user roles have access to the hosts contained within the affected host groups, or to the virtual machines deployed on those hosts.

### Parameters

#### **-JobGroup<Guid>**

Specifies an identifier for a series of commands that will run as a set just before the final command that includes the same job group identifier runs.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-ParentHostGroup<HostGroup>**

Specifies the parent host group that contains one or more hosts, host groups, or host clusters.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMHostGroup<HostGroup>**

Specifies a virtual machine host group object.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **HostGroup**

## **Notes**

- Requires a VMM host group object, which can be retrieved by using the Get-SCVMHostGroup cmdlet.

---

## Examples

### 1: Move one host group to a new parent host group.

The first command gets the host group named OldHostGroup and stores the host group object in the \$OldHostGroup variable.

The second command gets the host group named NewHostGroup and stores this host group object in the \$NewHostGroup variable.

The last command moves the host group stored in the \$OldHostGroup variable (OldHostGroup) to a location under its new parent host group, stored in the \$NewHostGroup variable.

```
PS C:\> $OldHostGroup = Get-SCVMHostGroup -Name "OldHostGroup"
```

```
PS C:\> $NewHostGroup = Get-SCVMHostGroup -Name "NewHostGroup"
```

```
PS C:\> Move-SCVMHostGroup -VMHostGroup $OldHostGroup -ParentHostGroup $NewHostGroup
```

### 2: Move all host groups to a new parent host group.

The first command gets all host group objects and stores them in the \$AllGroups array. This includes the default parent host group (All Hosts).

The second command creates a host group object named NewHostGroup01 and stores it in the \$HostGroup variable.

The last command passes each host group object stored in the \$AllGroups variable to the Move-VMHostGroup cmdlet, which moves each host group object to location stored in the \$HostGroup variable, except for All Hosts because All Hosts is the default parent host group and cannot be moved.

```
PS C:\> $AllGroups = Get-SCVMHostGroup
```

```
PS C:\> $HostGroup = New-SCVMHostGroup -Name "NewHostGroup01"
```

```
PS C:\> $AllGroups | Move-SCVMHostGroup -ParentHostGroup $HostGroup
```

## Related topics

[Get-SCVMHostGroup](#)

[New-SCVMHostGroup](#)

[Remove-SCVMHostGroup](#)

[Set-SCVMHostGroup](#)

---

# New-SCApplicationProfile

---

## New-SCApplicationProfile

Creates an application profile.

### Syntax

Parameter Set: Default

```
New-SCApplicationProfile [-Name] <String> [-ApplicationProfile <ApplicationProfile> ] [-CompatibilityType <String> ] [-Description <String> ] [-EnforceCompatibilityType] [-JobVariable <String> ] [-Owner <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-Tag <String> ] [-UserRole <UserRole> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The New-SCApplicationProfile cmdlet creates an application profile. Application profiles define the applications that will be installed during virtual machine deployment and servicing.

### Parameters

#### **-ApplicationProfile<ApplicationProfile>**

Specifies an application profile object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

#### **-CompatibilityType<String>**

Specifies the deployment types with which an application profile is compatible. Valid values are: General, SQLApplicationHost, WebApplicationHost.

---

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Description<String>**

States a description for the specified object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-EnforceCompatibilityType**

Indicates that artifacts from an application profile which is not compatible with the value provided for the CompatibilityType parameter are removed.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Owner<String>**

Specifies the owner of a VMM object in the form of a valid domain user account.

Example format: -Owner "Contoso\ReneeLo"

Example format: -Owner "ReneeLo@Contoso"

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Tag<String>**

Specifies a word or phrase to associate with an object so that you can search for all objects with the specified set of tags. You can search for a subset of tags, or you can search for the full set of tags.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false



---

Accept Wildcard Characters?	false
-----------------------------	-------

### **-UserRole<UserRole>**

Specifies a user role object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

### **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **ApplicationProfile**

---

## Examples

### 1: Create an application profile.

The first command creates an application profile object named SvcWebAppProfile01 and stores the object in the \$AppProfile variable.

The second displays information about the application profile stored in \$AppProfile to the user.

```
PS C:\> $AppProfile = New-SCApplicationProfile -Name "SvcWebAppProfile01" -Owner  
"Contoso\Katarina"  
PS C:\> $AppProfile
```

### 2: Create an application profile by cloning an existing profile.

The first command gets the application profile object named SvcWebAppProfile01 and stores the object in the \$AppProfile variable.

The second command creates an application profile named SvcWebAppProfile02 by cloning the application profile stored in \$AppProfile (SvcWebAppProfile01).

```
PS C:\> $AppProfile = Get-SCApplicationProfile -Name "SvcWebAppProfile01"  
PS C:\> $AppProfile02 = New-SCApplicationProfile -Name "SvcWebAppProfile02" -  
ApplicationProfile $AppProfile
```

## Related topics

[Get-SCApplicationProfile](#)

[Remove-SCApplicationProfile](#)

[Set-SCApplicationProfile](#)

---

# New-SCBaseline

---

## New-SCBaseline

Creates a baseline.

### Syntax

Parameter Set: Default

```
New-SCBaseline [-Name] <String> [-Description <String> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The New-SCBaseline cmdlet creates a baseline. A baseline is a list of updates which, together with scope assignments, can grade the compliance of required updates for Virtual Machine Manager (VMM) fabric servers.

### Parameters

#### **-Description<String>**

States a description for the specified object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **Baseline**

## **Examples**

### **1: Create a baseline.**

This command creates a new baseline named Security Baseline.

```
PS C:\> New-SCBaseline -Name "Security Baseline" -Description "Baseline that contains security updates"
```

---

## Related topics

[Get-SCBaseline](#)

[Remove-SCBaseline](#)

[Set-SCBaseline](#)

---

# New-SCCapabilityProfile

---

## New-SCCapabilityProfile

Creates a capability profile.

### Syntax

Parameter Set: FromExisting

```
New-SCCapabilityProfile -CapabilityProfile <CapabilityProfile> -Name <String> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: FromValues

```
New-SCCapabilityProfile -FabricCapabilityType {HyperV | ESX | Xen} -Name <String> [-CPUCompatibilityModeValue <Boolean> ] [-CPUCompatibilityModeValueCanChange <Boolean> ] [-CPUCountInitial <Int32> ] [-CPUCountMaximum <Int32> ] [-CPUCountMinimum <Int32> ] [-Description <String> ] [-DifferencingVirtualHardDiskValue <Boolean> ] [-DifferencingVirtualHardDiskValueCanChange <Boolean> ] [-DynamicMemoryValue <Boolean> ] [-DynamicMemoryValueCanChange <Boolean> ] [-DynamicVirtualHardDiskValue <Boolean> ] [-DynamicVirtualHardDiskValueCanChange <Boolean> ] [-ExistDiskStorageClassificationValue <Guid> ] [-FixedVirtualHardDiskValue <Boolean> ] [-FixedVirtualHardDiskValueCanChange <Boolean> ] [-JobVariable <String> ] [-LogicalNetworkValue <Guid> ] [-MaximumMemoryMBInitial <Int32> ] [-MaximumMemoryMBMaximum <Int32> ] [-MaximumMemoryMBMinimum <Int32> ] [-MemoryMBInitial <Int32> ] [-MemoryMBMaximum <Int32> ] [-MemoryMBMinimum <Int32> ] [-NetworkOptimizationValue <Boolean> ] [-NetworkOptimizationValueCanChange <Boolean> ] [-NewDiskStorageClassificationValue <Guid> ] [-OSCompatibilityModeValue <Boolean> ] [-OSCompatibilityModeValueCanChange <Boolean> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-SharedDVDImageFileValue <Boolean> ] [-SharedDVDImageFileValueCanChange <Boolean> ] [-StartupMemoryMBInitial <Int32> ] [-StartupMemoryMBMaximum <Int32> ] [-StartupMemoryMBMinimum <Int32> ] [-TargetMemoryBufferPercentInitial <Int32> ] [-TargetMemoryBufferPercentMaximum <Int32> ] [-TargetMemoryBufferPercentMinimum <Int32> ] [-VirtualDVDDriveCountInitial <Int32> ] [-VirtualDVDDriveCountMaximum <Int32> ] [-VirtualDVDDriveCountMinimum <Int32> ] [-VirtualHardDiskCountInitial <Int32> ] [-VirtualHardDiskCountMaximum <Int32> ] [-VirtualHardDiskCountMinimum <Int32> ] [-VirtualHardDiskSizeMBInitial <Int32> ] [-VirtualHardDiskSizeMBMaximum <Int32> ] [-VirtualHardDiskSizeMBMinimum <Int32> ] [-VirtualNetworkAdapterCountInitial <Int32> ] [-VirtualNetworkAdapterCountMaximum <Int32> ] [-VirtualNetworkAdapterCountMinimum <Int32> ] [-VMHighlyAvailableValue <Boolean> ] [-VMHighlyAvailableValueCanChange <Boolean> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

---

## Detailed Description

The New-SCCapabilityProfile cmdlet creates a capability profile object in Virtual Machine Manager (VMM). A capability profile is used to specify the capabilities of a virtual machine on a supported hypervisor when the virtual machine is deployed to a private cloud.

## Parameters

### **-CapabilityProfile<CapabilityProfile>**

Specifies a capability profile object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-CPUCompatibilityModeValue<Boolean>**

Indicates whether processor compatibility mode is enabled. When set to \$True, VMM limits the processor features that a virtual machine can use in order to improve compatibility with a different processor version.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-CPUCompatibilityModeValueCanChange<Boolean>**

Indicates whether the value for CPU compatibility mode can be updated.



---

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-CPUCountInitial<Int32>**

Specifies the initial number of processors that a virtual machine will have when deployed in a private cloud.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-CPUCountMaximum<Int32>**

Specifies the maximum number of processors that a virtual machine deployed in a private cloud can have.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

### **-CPUCountMinimum<Int32>**

Specifies the minimum number of processors that a virtual machine deployed in a private cloud can have.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Description<String>**

States a description for the specified object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-DifferencingVirtualHardDiskValue<Boolean>**

Indicates whether differencing disks are allowed.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-DifferencingVirtualHardDiskValueCanChange<Boolean>**

Indicates whether the value for differencing disks can be updated.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-DynamicMemoryValue<Boolean>**

Indicates whether dynamic memory is enabled.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-DynamicMemoryValueCanChange<Boolean>**

Indicates whether the value for dynamic memory can be updated.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-DynamicVirtualHardDiskValue<Boolean>**

Indicates whether dynamic virtual hard disks are allowed.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-DynamicVirtualHardDiskValueCanChange<Boolean>**

Indicates whether the value for dynamic virtual hard disks can be updated.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-ExistDiskStorageClassificationValue<Guid>**

Specifies a GUID.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-FabricCapabilityType<FabricCapabilityType>**

Specifies a fabric capability type. Fabric capability indicates the capabilities of the virtualization platform on which you will be deploying a virtual machine. VMM ensures that the settings in a capability profile are compatible with the selected fabric capability. Valid values are: HyperV, ESX, and Xen.

The acceptable values for this parameter are:

HyperV	
ESX	
Xen	

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-FixedVirtualHardDiskValue<Boolean>**

Indicates whether fixed virtual hard disks are allowed.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-FixedVirtualHardDiskValueCanChange<Boolean>**

Indicates whether the value for fixed virtual hard disks can be updated.

---

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-LogicalNetworkValue<Guid>**

Specifies a GUID.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-MaximumMemoryMBInitial<Int32>**

Specifies the initial maximum amount of memory, in megabytes (MB), allocated to a virtual machine when deployed in a private cloud.

---

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-MaximumMemoryMBMaximum<Int32>**

Specifies the highest amount of maximum memory, in megabytes (MB), that can be allocated to a virtual machine.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-MaximumMemoryMBMinimum<Int32>**

Specifies the lowest amount of maximum memory, in megabytes (MB), that can be allocated to a virtual machine.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-MemoryMBInitial<Int32>**

Specifies the initial amount of memory, in megabytes (MB), allocated to a virtual machine when deployed in a private cloud.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-MemoryMBMaximum<Int32>**

Specifies the maximum amount of memory, in megabytes (MB), that can be allocated to a virtual machine.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-MemoryMBMinimum<Int32>**

Specifies the minimum amount of memory, in megabytes (MB), that can be allocated to a virtual machine.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false



---

## **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-NetworkOptimizationValue<Boolean>**

Indicates whether network optimization is enabled.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-NetworkOptimizationValueCanChange<Boolean>**

Indicates whether the value for network optimization can be updated.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-NewDiskStorageClassificationValue<Guid>**

Specifies a GUID.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-OSCompatibilityModeValue<Boolean>**

Indicates whether operating system compatibility mode is enabled.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-OSCompatibilityModeValueCanChange<Boolean>**

Indicates whether the value for operating system compatibility can be updated.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-SharedDVDImageFileValue<Boolean>**

Indicates whether shared DVD image mode is enabled.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

### **-SharedDVDImageFileValueCanChange<Boolean>**

Indicates whether the value for shared DVD image mode can be updated.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-StartupMemoryMBInitial<Int32>**

Specifies the initial amount of memory, in megabytes (MB), that is allocated to a virtual machine upon startup.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-StartupMemoryMBMaximum<Int32>**

Specifies the maximum amount of memory, in megabytes (MB), that is allocated to a virtual machine upon startup.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false

---

Accept Wildcard Characters?	false
-----------------------------	-------

### **-StartupMemoryMBMinimum<Int32>**

Specifies the minimum amount of memory, in megabytes (MB), that is allocated to a virtual machine upon startup.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-TargetMemoryBufferPercentInitial<Int32>**

Specifies the initial percentage of memory above a virtual machine's current memory allocation that the host should try to reserve as a buffer.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-TargetMemoryBufferPercentMaximum<Int32>**

Specifies the maximum percentage of memory above a virtual machine's current memory allocation that the host should try to reserve as a buffer.

Aliases	none
Required?	false
Position?	named

---

Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-TargetMemoryBufferPercentMinimum<Int32>**

Specifies the minimum percentage of memory above a virtual machine's current memory allocation that the host should try to reserve as a buffer.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VirtualDVDDriveCountInitial<Int32>**

Specifies the initial number of DVD drives attached to a virtual machine when deployed in a private cloud.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VirtualDVDDriveCountMaximum<Int32>**

Specifies the maximum number of DVD drives that can be attached to a virtual machine deployed in a private cloud.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VirtualDVDDriveCountMinimum<Int32>**

Specifies the minimum number of DVD drives that can be attached to a virtual machine deployed in a private cloud.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VirtualHardDiskCountInitial<Int32>**

Specifies the initial number of virtual hard disks attached to a virtual machine when deployed in a private cloud.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VirtualHardDiskCountMaximum<Int32>**

Specifies the maximum number of virtual hard disks that can be attached to a virtual machine deployed in a private cloud.

---

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VirtualHardDiskCountMinimum<Int32>**

Specifies the minimum number of virtual hard disks that can be attached to a virtual machine deployed in a private cloud.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VirtualHardDiskSizeMBInitial<Int32>**

Specifies the initial hard disk size, in megabytes (MB), for a virtual machine when deployed in a private cloud.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false



---

### **-VirtualHardDiskSizeMBMaximum<Int32>**

Specifies the maximum virtual hard disk size, in megabytes (MB), allowed for a virtual machine deployed in a private cloud.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VirtualHardDiskSizeMBMinimum<Int32>**

Specifies the minimum virtual hard disk size, in megabytes (MB), allowed for a virtual machine deployed in a private cloud.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VirtualNetworkAdapterCountInitial<Int32>**

Specifies the initial number of virtual network adapters attached to a virtual machine when deployed in a private cloud.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

### **-VirtualNetworkAdapterCountMaximum<Int32>**

Specifies the maximum number of virtual network adapters that can be attached to a virtual machine deployed in a private cloud.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VirtualNetworkAdapterCountMinimum<Int32>**

Specifies the minimum number of virtual network adapters that can be attached to a virtual machine deployed in a private cloud.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMHighlyAvailableValue<Boolean>**

Indicates whether a deployed virtual machine will be highly available.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false

Accept Wildcard Characters?	false
-----------------------------	-------

### **-VMHighlyAvailableValueCanChange<Boolean>**

Indicates whether the value indicating the high availability status of a virtual machine can be updated.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

### **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **CloudCapabilityProfile**

---

## Examples

### 1: Create a capability profile that is compatible with Hyper-V hosts.

The first command creates a capability profile object named CapabilityProf01 that is compatible with Hyper-V hosts and stores the object in the \$CapabilityProfile variable.

The second command displays information about the capability profile stored in \$CapabilityProfile to the user.

```
PS C:\> $CapabilityProfile = New-SCCapabilityProfile -Name "CapabilityProf01" -  
FabricCapabilityType "HyperV"  
PS C:\> $CapabilityProfile
```

## Related topics

[Get-SCCapabilityProfile](#)

[Remove-SCCapabilityProfile](#)

[Set-SCCapabilityProfile](#)

[Test-SCCapabilityProfile](#)

---

# New-SCCloud

---

## New-SCCloud

Creates a private cloud.

### Syntax

Parameter Set: VMHostGroup

```
New-SCCloud [-Name] <String> -VMHostGroup <HostGroup[]> [-Description <String> ] [-JobGroup <Guid> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: VMwareResourcePool

```
New-SCCloud [-Name] <String> -VMwareResourcePool <VmwareResourcePool> [-Description <String> ] [-JobGroup <Guid> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The New-SCCloud cmdlet creates a private cloud in Virtual Machine Manager (VMM). A private cloud is a cloud that is provisioned and managed on-premise by an organization. The private cloud is deployed using an organization's own hardware to leverage the advantages of the private cloud model. Through VMM, an organization can manage the private cloud definition, access to the private cloud, and the underlying physical resources.

You can create a private cloud from the following resources:

- Host groups that contain resources from Hyper-V hosts, Citrix XenServer hosts, and VMware ESX hosts
- A VMware resource pool

For more information about private clouds, see "Creating a Private Cloud Overview" in the TechNet library at <http://go.microsoft.com/fwlink/?LinkID=212407>.

For information about private cloud capacity, type: "Get-Help Set-SCCloudCapacity -detailed."

### Parameters

#### **-Description<String>**

States a description for the specified object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-JobGroup<Guid>**

Specifies an identifier for a series of commands that will run as a set just before the final command that includes the same job group identifier runs.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Name<String>**

Specifies the name of a VMM object.

---

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMHostGroup<HostGroup[]>**

Specifies a virtual machine host group object or an array of host group objects.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-VMwareResourcePool<VmwResourcePool>**

Assigns a virtual machine deployed on a VMware ESX host or a private cloud to a specific VMware resource pool.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false



---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **Cloud**

## Examples

### 1: Create a private cloud from a host group.

The first command gets the host group named HostGroup01 and stores it in the \$HostGroup variable.

The second command creates a private cloud named Cloud01 from the host group stored in the \$HostGroup variable.

```
PS C:\> $HostGroup = Get-SCVMHostGroup -Name "HostGroup01"
```

```
PS C:\> New-SCCloud -Name "Cloud01" -VMHostGroup $HostGroup
```

### 2: Create a private cloud using a job group

The first command creates a new GUID and stores it in the \$Guid variable.

The second command creates a job group using the GUID stored in \$Guid.

The third command gets the host group object named HostGroup02 and stores the object in the \$HostGroup variable.

The last command creates a private cloud named Cloud02, using the job group created in the second command and HostGroup02 for its resources.

```
PS C:\> $Guid = [System.Guid]::NewGuid()
```

```
PS C:\> Set-SCCloud -JobGroup $Guid
```

```
PS C:\> $HostGroup = Get-SCVMHostGroup -Name "HostGroup02"
```

```
PS C:\> New-SCCloud -JobGroup $Guid -Name "Cloud02" -VMHostGroup $HostGroup -Description  
"This is a cloud for HostGorup02"
```

### 3: Create a private cloud from multiple host groups.

The first command creates an object array named \$HostGroups. The second and third commands populate the object array with the host groups named Seattle and New York.

The last command creates a private cloud named Cloud03 using the host groups stored in the \$HostGroups array as its resources.

```
PS C:\> $HostGroups = @()
```

```
PS C:\> $HostGroups += Get-SCVMHostGroup -Name "Seattle"
```

---

```
PS C:\> $HostGroups += Get-SCVMHostGroup -Name "New York"
```

```
PS C:\> New-SCCloud -VMHostGroup $HostGroups -Name "Cloud03" -Description "Cloud for the  
Seattle and New York host groups"
```

## Related topics

[Get-SCCloud](#)

[Remove-SCCloud](#)

[Set-SCCloud](#)

---

# New-SCComputerConfiguration

---

## New-SCComputerConfiguration

Creates a computer configuration object by gathering computer configuration information from a physical source machine that you plan to convert to a virtual machine managed by VMM.

### Syntax

Parameter Set: Default

```
New-SCComputerConfiguration -Credential <PSCredential> -SourceComputerName <String> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The New-SCComputerConfiguration cmdlet creates a computer configuration object by gathering computer configuration information from a physical machine that you plan to convert to a virtual machine managed by Virtual Machine Manager (VMM). Information about a computer's hardware, physical disks, and operating system is stored in the machine configuration object.

New-SCComputerConfiguration installs the VMM P2V agent software on the physical source machine, runs the configuration information gathering process, and creates and stores the resulting machine configuration object (which is associated with this physical source computer) in the VMM database.

A computer configuration is used when you use the New-SCP2V cmdlet to convert a physical machine to a virtual machine. To perform this conversion, you use a physical computer as a model from which to create an identical, or nearly identical, virtual machine that has the same identity (ComputerName.DomainName) as the physical machine.

For information about the operating systems for which VMM supports P2V, see "Converting Physical Computers to Virtual Machines in VMM" in the TechNet Library at <http://go.microsoft.com/fwlink/?LinkId=225101>.

### Parameters

#### **-Credential<PSCredential>**

Specifies a credential object or, for some cmdlets, a Run As account object that contains the user name and password of an account that has permission to perform this action. Or, in the case of Restart-SCJob, has permission to complete a restarted task.

For more information about the PSCredential object, type: "Get-Help Get-Credential".

For more information about Run As accounts, type: "Get-Help New-SCRunAsAccount".

---

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-SourceComputerName<String>**

Specifies the source computer for a physical-to-virtual (P2V) machine conversion performed by VMM. Valid formats: FQDN, IPv4 or IPv6 address, or NetBIOS name.

Note: See the examples for a specific cmdlet to determine how that cmdlet specifies the source computer name.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **VMConfiguration**

## Examples

### 1: Gather information from a physical source machine.

The first command uses Get-Credential to prompt you to supply a user name and password and stores your credentials in the \$Credential variable. The required credentials for this operation are either a local Administrator account or a domain account with administrator rights on the computer from which you want to gather information.

The second gathers the computer configuration information from the physical source machine called P2VSource01 in the Contoso.com domain. As this command is processed, \$Credential provides your credentials to New-SCComputerConfiguration. The New-ComputerConfiguration cmdlet stores the resulting machine configuration object associated with P2VSource01.Contoso.com in the VMM database.

```
PS C:\> $Credential = Get-Credential
PS C:\> New-SCComputerConfiguration -SourceComputerName "P2VSource01.Contoso.com" -
Credential $Credential
```

### 2: Determine the required patches for a particular conversion.

The first command uses Get-Credential to prompt you to supply a user name and password and stores your credentials in the \$Credential variable. The required credentials for this operation are either a local Administrator account or a domain account with administrator rights on the computer from which you want to gather information.

The second gathers the computer configuration information from the physical source machine named P2VSource02 in the Contoso.com domain, and it stores the information in \$ComputerConfig. As this command is processed, \$Credential provides your credentials to New-SCComputerConfiguration.

The last command displays the list of errors, if any, that were detected on the source machine. Any items with the value 'Error' must be resolved before you attempt a physical-to-virtual conversion.

```
PS C:\> $Credential = Get-Credential
PS C:\> $ComputerConfig = New-SCComputerConfiguration -SourceComputerName
"P2VSource02.Contoso.com" -Credential $Credential
PS C:\> $ComputerConfig.ErrorList
```

---

## Related topics

[Add-SCPatch](#)

[Get-SCComputerConfiguration](#)

[New-SCP2V](#)

[Remove-SCComputerConfiguration](#)

---

# New-SCCustomProperty

---

## New-SCCustomProperty

Creates a custom property definition in the VMM database.

### Syntax

Parameter Set: Default

```
New-SCCustomProperty [-Name] <String> -AddMember {VM | VMHost | Template | ServiceTemplate | ServiceInstance | Cloud | VMHostGroup | HostCluster | ComputerTier} [-Description <String> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The New-SCCustomProperty cmdlet creates a custom property definition in the Virtual Machine Manager (VMM) database.

You can create a custom property for the following object types: VM, Template, VMHost, HostCluster, VMHostGroup, ServiceTemplate, ServiceInstance, ComputerTier, Cloud.

### Parameters

#### **-AddMember<CustomPropertyObjectType[]>**

Adds one or more members to an object that has the concept of members, such as a group. For example, AddMember adds one or more Active Directory domain users or groups to a user role.

Example formats:

-AddMember Domain\User

-AddMember User

-AddMember User@Domain

-AddMember Domain\LabGroupAlias

-AddMember LabGroupAlias (an Active Directory security group, not an email alias)

The acceptable values for this parameter are:

VM	
VMHost	



Template	
ServiceTemplate	
ServiceInstance	
Cloud	
VMHostGroup	
HostCluster	
ComputerTier	

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

**-Description<String>**

States a description for the specified object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

**-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## **Examples**

### **1: Create a custom property.**

The first command creates a custom property object named Cost Center with VM as a member and stores the object in the \$CustomProp variable.

The second command displays the properties of the custom property object stored in \$CustomProp to the user.

```
PS C:\> $CustomProp = New-SCCustomProperty -Name "Cost Center" -AddMember "VM"
PS C:\> $CustomProp
```

## **Related topics**

[Get-SCCustomProperty](#)

---

[Remove-SCCustomProperty](#)

[Set-SCCustomProperty](#)

---

# New-SCDefaultGateway

---

## New-SCDefaultGateway

Creates a default gateway object that is used when creating or modifying static IP address pools.

### Syntax

Parameter Set: Automatic

```
New-SCDefaultGateway -IPAddress <String> [-Automatic] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ByMetric

```
New-SCDefaultGateway -IPAddress <String> -Metric <Int32> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The New-SCDefaultGateway cmdlet creates a Virtual Machine Manager (VMM) default gateway object that is used when creating or modifying static IP address pools. The default metric is automatic; to change this setting, use the Metric parameter.

### Parameters

#### -Automatic

Indicates if the metric associated with a network gateway is automatically computed.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-IPAddress<String>**

Specifies an IPv4 or IPv6 address.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Metric<Int32>**

Specifies a numerical metric associated with a network gateway.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **DefaultGateway**

## Examples

### 1: Create a default gateway object with an automatically calculated metric.

This command creates a gateway object with an IP address of 10.0.0.1, automatically computes the metric for the gateway object, and then stores the object in the \$Gateway variable.

```
PS C:\> $Gateway = New-SCDefaultGateway -IPAddress 10.0.0.1 -Automatic
```

### 2: Create a default gateway object and manually set its metric.

This command creates a gateway object with an IP address of 10.0.0.1, sets its metric to 10, and then stores the object in the \$Gateway variable.

```
PS C:\> $Gateway = New-SCDefaultGateway -IPAddress 10.0.1.1 -Metric 10
```

---

# New-SCExternalJob

---

## New-SCExternalJob

Creates a new external job in VMM.

### Syntax

Parameter Set: Default

```
New-SCExternalJob [-Name] <String> [-ResultObject <ClientObject> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The New-SCExternalJob cmdlet creates external jobs in Virtual Machine Manager (VMM).

External jobs add non-VMM jobs to the jobs view in the VMM console. These jobs are typically used to allow other applications integrating with VMM to serve status updates into the console so that users of the console can see the updates. For example, if you create an Add-in, you might use external jobs to provide the status of work completed on another system.

### Parameters

#### **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

#### **-ResultObject<ClientObject>**

Specifies the object to which a new external job is related.



Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **Task**

## **Examples**

### **1: Create an external job.**

The first command gets the virtual machine object named VM01 and stores the object in the \$VMMObject variable. The object in this command can be any VMM object for which you are creating a job.

The second command creates an external job named ExternalJob01 and relates the job to VM01 which is stored in the \$VMMObject variable.

---

```
PS C:\> $VMMObject = Get-SCVirtualMachine -Name "VM01"
```

```
PS C:\> New-SCExternalJob -Name "ExternalJob01" -ResultObject $VMMObject
```

## Related topics

[Set-SCExternalJob](#)

---

# New-SCGuestOSProfile

---

## New-SCGuestOSProfile

Creates a guest operating system profile for use in VMM.

### Syntax

Parameter Set: Default

```
New-SCGuestOSProfile [-Name] <String> [-AnswerFile <Script> ] [-AutoLogonCount <UInt32> ] [-AutoLogonCredential <RunAsAccount> ] [-ComputerName <String> ] [-Description <String> ] [-DisableAutoLogon] [-DomainJoinOrganizationalUnit <String> ] [-FullName <String> ] [-GuestOSProfile <GuestOSProfile> ] [-GuiRunOnceCommands <String[]> ] [-JobVariable <String> ] [-LinuxAdministratorSSHKey <SSHKey> ] [-LinuxDomainName <String> ] [-LocalAdministratorCredential <VMMSecret> ] [-MergeAnswerFile <Boolean> ] [-OperatingSystem <OperatingSystem> ] [-OrganizationName <String> ] [-Owner <String> ] [-ProductKey <String> ] [-PROTipID <Guid> ] [-RemoveDomainJoinOrganizationalUnit] [-RemoveServerFeatures] [-RunAsynchronously] [-TimeZone <Int32> ] [-UnattendSettings <Dictionary<String>> ] [-UserRole <UserRole> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: Domain

```
New-SCGuestOSProfile [-Name] <String> -Domain <String> [-AnswerFile <Script> ] [-AutoLogonCount <UInt32> ] [-AutoLogonCredential <RunAsAccount> ] [-ComputerName <String> ] [-Description <String> ] [-DisableAutoLogon] [-DomainJoinCredential <VMMSecret> ] [-DomainJoinOrganizationalUnit <String> ] [-FullName <String> ] [-GuestOSProfile <GuestOSProfile> ] [-GuiRunOnceCommands <String[]> ] [-JobVariable <String> ] [-LinuxAdministratorSSHKey <SSHKey> ] [-LinuxDomainName <String> ] [-LocalAdministratorCredential <VMMSecret> ] [-MergeAnswerFile <Boolean> ] [-OperatingSystem <OperatingSystem> ] [-OrganizationName <String> ] [-Owner <String> ] [-ProductKey <String> ] [-PROTipID <Guid> ] [-RemoveDomainJoinOrganizationalUnit] [-RemoveServerFeatures] [-RunAsynchronously] [-TimeZone <Int32> ] [-UnattendSettings <Dictionary<String>> ] [-UserRole <UserRole> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: Workgroup

```
New-SCGuestOSProfile [-Name] <String> [-AnswerFile <Script> ] [-AutoLogonCount <UInt32> ] [-AutoLogonCredential <RunAsAccount> ] [-ComputerName <String> ] [-Description <String> ] [-DisableAutoLogon] [-DomainJoinOrganizationalUnit <String> ] [-FullName <String> ] [-GuestOSProfile <GuestOSProfile> ] [-GuiRunOnceCommands <String[]> ] [-JobVariable <String> ] [-LinuxAdministratorSSHKey <SSHKey> ] [-LinuxDomainName <String> ] [-LocalAdministratorCredential <VMMSecret> ] [-MergeAnswerFile <Boolean> ] [-OperatingSystem <OperatingSystem> ] [-OrganizationName <String> ] [-Owner <String> ] [-ProductKey <String> ] [-PROTipID <Guid> ] [-RemoveDomainJoinOrganizationalUnit] [-RemoveServerFeatures] [-RunAsynchronously] [-TimeZone <Int32> ] [-UnattendSettings <Dictionary<String>> ] [-UserRole <UserRole> ] [-VMMServer <ServerConnection> ] [-Workgroup <String> ] [ <CommonParameters>]
```

---

## Detailed Description

The New-SCGuestOSProfile cmdlet creates a guest operating system profile for use in Virtual Machine Manager (VMM). A guest operating system is the operating system on a virtual machine, in contrast to a host operating system which is on the physical host computer on which one or more virtual machines are deployed.

A guest operating system profile stores operating system configuration information. A profile consists of a set of properties containing the most common settings specified in an unattended answer file (such as a Sysprep.inf or an Unattend.xml file).

You can create a standalone guest operating system profile or customize a template or virtual machine to include guest operating system profile settings. The New-SCGuestOSProfile cmdlet stores the new guest operating system profile object in the VMM library.

You can create a guest operating system profile based on default settings, based on an existing guest operating system profile, or you can customize settings as you create the profile. If you specify no parameters except the Name parameter, the New-SCGuestOSProfile cmdlet creates a default guest operating system profile object.

## Parameters

### **-AnswerFile<Script>**

Specifies a script object stored in the VMM library to use as an answer file. The name of the answer file script depends on the operating system that you want to install on a virtual machine:

ANSWER FILE    GUEST OS TO INSTALL ON VM

-----    -----

Sysprep.inf    Windows XP, Windows Server 2000, or Windows Server 2003

Unattend.xml    Windows Vista, Windows 7, or Windows Server 2008

Aliases	SysPrepFile
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

---

## **-AutoLogonCount<UInt32>**

Specifies the number of times that Windows should automatically log the administrator denoted in the Answer File on to the console session.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-AutoLogonCredential<RunAsAccount>**

Specifies the Run As account contained in the Answer File that Windows uses to log on to the console session when automatic administrator logon is enabled.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-ComputerName<String>**

Specifies the name of a computer that VMM can uniquely identify on your network. Valid formats are: FQDN, IPv4 or IPv6 address, or NetBIOS name.

NOTE: See the examples for a specific cmdlet to determine how that cmdlet specifies the computer name.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Description<String>**

States a description for the specified object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-DisableAutoLogon**

Disables automatic administrator logon.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Domain<String>**

Specifies a fully qualified domain name (FQDN) for an Active Directory domain.

Example format: -Domain "Domain01.Corp.Contoso.com"

Aliases	none
Required?	true
Position?	named

Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-DomainJoinCredential<VMMCredential>**

Specifies the user name and password of an account with permission to join a computer to the domain. A limited rights account should be used for joining computers (either physical or virtual) to the domain.

Example format for a PS credential:

```
$DomainJoinCredential = Get-Credentital
```

```
-Domain "ThisDomain.Corp.Contoso.com" -DomainJoinCredential $DomainJoinCredential
```

Example format for a Run As account:

```
$DomainJoinCredential = Get-SCRunAsAccount -Name "RunAsAcct01"
```

```
-Domain "ThisDomain.Corp.Contoso.com" -DomainJoinCredential $DomainJoinCredential
```

NOTE: You can use the DomainJoinCredential parameter to specify credentials (on a VMHostProfile) for joining a physical host computer to the domain, or to specify credentials (on a new or existing template, on a new or existing guest operating system profile, or on a new virtual machine) for joining a virtual machine to the domain.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-DomainJoinOrganizationalUnit<String>**

Specifies the organizational unit to which the computer is joined during an unattended mini-setup.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false

Accept Wildcard Characters?	false
-----------------------------	-------

### **-FullName<String>**

Specifies the name of the person in whose name a virtual machine is registered.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-GuestOSProfile<GuestOSProfile>**

Specifies a guest operating system profile object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-GuiRunOnceCommands<String[]>**

Specifies one or more commands to add to the [GuiRunOnce] section of an unattended answer file (such as Unattend.xml). Use single quotes around each string enclosed in double quotes.

Example format:

```
-GuiRunOnceCommands "C:\APF\APFPostSysPrepCopy.cmd PARAMS1",
"C:\APF\APFPostSysPrepCopy.cmd PARAMS1"
```

For information about how Windows PowerShell uses quotes, type: "Get-Help about\_Quoting\_Rules"

Aliases	none
---------	------



---

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-LinuxAdministratorSSHKey<SSHKey>**

Specifies the public key file for a Linux Ssh Key.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-LinuxDomainName<String>**

Specifies a fully qualified domain name (FQDN) to be used in conjunction with Linux operating system specialization.

Example format: -LinuxDomainName "Domain01.Corp.Contoso.com"

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

**-LocalAdministratorCredential<VMMSecret>**

Specifies the user name and password for the Local Administrator account (or Linux root account in the case of a Linux compatible Guest Operating System profile).

Specifying credentials on a new or existing template, on a new or existing guest operating system profile, or on a new virtual machine overrides any existing Administrator password.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

**-MergeAnswerFile<Boolean>**

Specifies that the cmdlet merge the specified answer file with the specified guest operating system settings. The default value is TRUE. This parameter is used by the VMM console. You do not need to use this parameter at the command prompt.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-OperatingSystem<OperatingSystem>**

Specifies the type of operating system for a virtual machine. To list the names of all available operating systems in VMM, type: "Get-SCOperatingSystem".

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-OrganizationName<String>**

Specifies the name of the organization for the person in whose name a virtual machine is registered.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-Owner<String>**

Specifies the owner of a VMM object in the form of a valid domain user account.

Example format: -Owner "Contoso\ReneeLo"

Example format: -Owner "ReneeLo@Contoso"

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-ProductKey<String>**

Specifies a product key. The product key is a 25-digit number that identifies the product license. A product key can be used to register VMM or an operating system to be installed on a virtual machine or host.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false

---

Accept Wildcard Characters?	false
-----------------------------	-------

### **-RemoveDomainJoinOrganizationalUnit**

Removes the organizational unit that the computer joined during setup.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-RemoveServerFeatures**

Removes all server roles and features from a template.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false

---

Accept Wildcard Characters?	false
-----------------------------	-------

### **-TimeZone<Int32>**

Specifies a number (an index) that identifies a geographical region that shares the same standard time. For a list of time zone indexes, see "Microsoft Time Zone Index Values" at: <http://go.microsoft.com/fwlink/?LinkId=120935>. If no time zone is specified, the default time zone used for a virtual machine is the same time zone setting that is on the virtual machine host.

Example format to specify the GMT Standard Time zone: -TimeZone 085

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-UnattendSettings<Dictionary<String>>**

Specifies a key/value pair consisting of string, string.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-UserRole<UserRole>**

Specifies a user role object.

Aliases	none
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-Workgroup<String>**

Specifies on a new or existing template, on a new or existing guest operating system profile, or on a new virtual machine the name of the workgroup to which you want to join a virtual machine. You can use this parameter to override the existing value on a template or on a guest operating system profile.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

---

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **GuestOSProfile**

## Examples

### 1: Create a default guest operating system profile.

The first command gets the operating system object named "64-bit edition of Windows Server 2008 R2 Enterprise" from VMMServer01 and stores the object in the \$OS variable.

The second command creates a default guest operating system profile named NewProfile01 and specifies the name of the operating system.

```
PS C:\> $OS = Get-SCOperatingSystem -VMMServer "VMMServer01.Contoso.com" | where {$_.Name - eq "64-bit edition of Windows Server 2008 R2 Enterprise"}
```

```
PS C:\> New-SCGuestOSProfile -Name "NewOSProfile01" -OperatingSystem $OS
```

### 2: Create a guest operating system profile to use to join a virtual machine to a domain.

The first command gets the Run As account object named RunAsAccount01 and stores the object in the \$DomainCredential variable.

The second command gets the operating system object named "64-bit edition of Windows Server 2008 R2 Enterprise" from VMMServer01 and stores the object in the \$OS variable.

The last command creates a guest operating system profile called NewOSProfile02 and specifies the name of the operating system. When you use NewOSProfile02 to create a virtual machine, the profile will use the specified Run As account to join the virtual machine to the Contoso.com domain.

NOTE: If you provide a PSCredential object instead of a RunAs account for the DomainJoinCredential parameter, it is recommended practice to use only an account with limited rights for automatically joining a virtual machine to a domain because the domain account name and password appear briefly in plain text on the host server.

```
PS C:\> $DomainCredential = Get-SCRunAsAccount -Name "RunAsAccount01"
```

```
PS C:\> $OS = Get-SCOperatingSystem -VMMServer "VMMServer01.Contoso.com" | where {$_.Name - eq "64-bit edition of Windows Server 2008 R2 Enterprise"}
```

```
PS C:\> New-SCGuestOSProfile -Name "NewOSProfile02" -JoinDomain "Contoso.com" - DomainJoinCredential $DomainCredential -OperatingSystem $OS
```

### 3: Clone an existing guest operating system profile.

The first command gets the guest OS profile object named NewOSProfile02 and stores the object in the \$OSProfile variable.



---

The second command gets the operating system object named "64-bit edition of Windows Server 2008 R2 Datacenter" and stores the object in the \$OS variable.

The last command creates guest operating system profile named NewOSProfile03, which is based on NewOSProfile02, but modifies the computer name and specifies a different name for the operating system. All other settings in NewOSProfile03 are identical to those in NewOSProfile02.

```
PS C:\> $OSProfile = Get-SCGuestOSProfile -Name "NewOSProfile02"
```

```
PS C:\> $OS = Get-SCOperatingSystem -VMMServer "VMMServer01.contoso.com" | where {$_.Name - eq "64-bit edition of Windows Server 2008 R2 Datacenter"}
```

```
PS C:\> New-SCGuestOSProfile -Name "NewOSProfile03" -GuestOSProfile $OSProfile -ComputerName "Contoso03" -OperatingSystem $OS
```

## 4: Create a Linux guest operating system profile and set the Linux DNS domain name

The first command gets the operating system object named "CentOS Linux 6 (64 bit)" from VMMServer01 and stores the object in the \$OS variable.

The second command creates a guest operating system profile called "My CentOS Profile" and specifies the name of the operating system and the Linux DNS domain name. When you use the My CentOS Profile to create a virtual machine, the profile will use the specified Linux DNS domain name.

```
PS C:\> $OS = Get-SCOperatingSystem -VMMServer "VMMServer01.Contoso.com" | where {$_.Name - eq "CentOS Linux 6 (64 bit)"}
```

```
PS C:\> New-SCGuestOSProfile -Name "My CentOS Profile" -LinuxDomainName "contoso.com" - OperatingSystem $os
```

### Related topics

[Get-SCGuestOSProfile](#)

[Get-SCVMMServer](#)

[New-SCVirtualMachine](#)

[New-SCVMTemplate](#)

[Remove-SCGuestOSProfile](#)

[Set-SCGuestOSProfile](#)

[Set-SCVMTemplate](#)

---

# New-SCHardwareProfile

---

## New-SCHardwareProfile

Creates a hardware profile in the VMM library.

### Syntax

Parameter Set: Default

```
New-SCHardwareProfile [-Name] <String> [-BootOrder {Floppy | CD | IdeHardDrive | PxeBoot | Unknown} ] [-CapabilityProfile <CapabilityProfile> ] [-CPUCount <Byte> ] [-CPUExpectedUtilizationPercent <Int32> ] [-CPULimitForMigration <Boolean> ] [-CPULimitFunctionality <Boolean> ] [-CPUMaximumPercent <Int32> ] [-CPUPerVirtualNumaNodeMaximum <Byte> ] [-CPURelativeWeight <Int32> ] [-CPUReserve <Int32> ] [-CPUType <ProcessorType> ] [-Description <String> ] [-DiskIops <Int32> ] [-DRProtectionRequired <Boolean> ] [-DynamicMemoryBufferPercentage <Int32> ] [-DynamicMemoryEnabled <Boolean> ] [-DynamicMemoryMaximumMB <Int32> ] [-DynamicMemoryMinimumMB <UInt16> ] [-HardwareProfile <HardwareProfile> ] [-HAVMPriority <UInt32> ] [-HighlyAvailable <Boolean> ] [-JobGroup <Guid> ] [-JobVariable <String> ] [-MemoryMB <Int32> ] [-MemoryPerVirtualNumaNodeMaximumMB <Int32> ] [-MemoryWeight <Int32> ] [-MonitorMaximumCount <Int32> ] [-MonitorMaximumResolution <String> ] [-NetworkUtilizationMbps <Int32> ] [-NumaIsolationRequired <Boolean> ] [-NumLock <Boolean> ] [-Owner <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-UserRole <UserRole> ] [-VirtualNumaNodesPerSocketMaximum <Byte> ] [-VirtualVideoAdapterEnabled <Boolean> ] [-VMMServer <ServerConnection> ] [  
<CommonParameters>]
```

### Detailed Description

The New-SCHardwareProfile cmdlet creates a hardware profile for use in Virtual Machine Manager (VMM) that stores hardware configuration information. You can create a standalone hardware profile or customize a template or virtual machine to include hardware profile settings. New-SCHardwareProfile stores the new hardware profile object in the VMM library.

You can create a hardware profile based on defaults or an existing hardware profile, or, you can customize a hardware profile as you create it. If you specify no parameters except the Name parameter (which is required), VMM creates a default hardware profile object.

Hardware profile settings that you can configure for a virtual machine include:

- Boot order settings in the BIOS that specify the device startup order for a virtual machine.

NOTE: The boot order setting is available only for virtual machines on a Hyper-V host or Citrix XenServer host.

- CPU settings for a virtual machine.

- Memory available on a virtual machine.
- A virtual floppy drive.
- Two virtual COM ports (COM1 and COM2).
- A built-in virtual IDE device.
- One or more virtual SCSI adapters.
- One or more virtual network adapters that you can attach to a logical network. A virtual network adapter can be emulated or synthetic.
- The priority assigned to a virtual machine for using the host's CPU resources relative to the use of the host's CPU by other virtual machines deployed on the same host. CPU priorities are determined by the virtualization software.
- Whether a virtual machine created from this profile will be highly available. A highly available virtual machine is a virtual machine that can only be placed on a host that is part of a host cluster.

## Parameters

### **-BootOrder<BootDevice[]>**

Specifies the order of devices that a virtual machine on a Hyper-V host uses to start up. Valid values are: CD, IDEHardDrive, PXEBoot, Floppy.

Example format: -BootOrder PXEBoot,IDEHardDrive,CD,Floppy

The acceptable values for this parameter are:

Floppy	
CD	
IdeHardDrive	
PxeBoot	
Unknown	

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-CapabilityProfile<CapabilityProfile>**

Specifies a capability profile object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-CPUCount<Byte>**

Specifies the number of CPUs on a virtual machine, on a hardware profile, or on a template. See the examples for a specific cmdlet to determine how that cmdlet uses this parameter.

TYPE OF HOST    NUMBER OF PROCESSORS

-----    -----

Hyper-V        Up to 4 CPUs per VM; varies by guest OS

VMware ESX    Up to 4 CPUs per VM for any supported guest OS

Exception: 1 CPU on a VM running Windows NT 4.0

Citrix XenServer Up to 8 CPUs per VM; varies by guest OS

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-CPUExpectedUtilizationPercent<Int32>**

Specifies the percent of CPU on the host that you expect this virtual machine to use. This value is used only when VMM determines a suitable host for the virtual machine.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-CPULimitForMigration<Boolean>**

Limits, when set to \$True, processor features for the specified virtual machine in order to enable migration to a physical computer that has a different version of the same processor as the source computer. VMM does not support migrating virtual machines between physical computers that have processors from different manufacturers.

Aliases	LimitCPUForMigration
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-CPULimitFunctionality<Boolean>**

Enables running an older operating system (such as Windows NT 4.0) on a virtual machine deployed on a Hyper-V host or on a VMware ESX host by providing limited CPU functionality for the virtual machine.

Aliases	LimitCPUFunctionality
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-CPUMaximumPercent<Int32>**

Specifies the highest percentage of the total resources of a single CPU on the host that can be used by a specific virtual machine at any given time.

Example: -CPUMaximumPercent 80 (to specify 80 per cent)

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-CPUPerVirtualNumaNodeMaximum<Byte>**

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-CPURelativeWeight<Int32>**

Specifies the amount of CPU resources on a host that this virtual machine can use relative to other virtual machines on the same host. A virtual machine with a higher setting is allocated more CPU resources than a virtual machine with a lower setting.

TYPE OF HOST    RANGE OF RELATIVE VALUES

-----    -----

Hyper-V        1 to 10000

VMware ESX    2000 = High

---

1500 = Above Normal  
1000 = Normal (default)  
750 = Below Normal  
500 = Low  
1 to 1000000 = Custom

The VMware term for these values is "shares."

Citrix XenServer 1 to 65536, normal is 256

NOTE: See the examples for a specific cmdlet to determine how that cmdlet uses this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-CPUReserve<Int32>**

Specifies the minimum percentage of the resources of a single CPU on the host to allocate to a virtual machine. The percentage of CPU capacity that is available to the virtual machine is never less than this percentage.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-CPUType<ProcessorType>**

Specifies the type of CPU for a virtual machine. To retrieve a list of all CPU types that are available for use in virtual machines in a VMM environment, type: "Get-SCCPUType"

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Description<String>**

States a description for the specified object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-DiskIops<Int32>**

Specifies the number of disk input/output operations per second (IOPS) on the host that can be used by a specific virtual machine.

Example: -DiskIO 1500 (to specify 1500 IOPS).

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-DRProtectionRequired<Boolean>**

This parameter is reserved for future use.



Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-DynamicMemoryBufferPercentage<Int32>**

Specifies the percentage of memory above a virtual machine's current memory allocation which the host should try to reserve as a buffer. The default value is 20.

Example format: -DynamicMemoryTargetBufferPercentage 20

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-DynamicMemoryEnabled<Boolean>**

Enables, when set to \$True, dynamic memory for virtual machines. You can enable dynamic memory directly on a virtual machine, or on a template or hardware profile that will be used to create virtual machines. The default value is False.

REQUIRED: You can enable dynamic memory for a virtual machine only if that virtual machine is deployed on a host running Windows Server 2008 SP1 or later or if the virtual machine is stored in a library in a stopped state (hardware changes to a stored virtual machine can only be made if the virtual machine does not have snapshots). Enabling dynamic memory on a virtual machine stored in a library will limit placement of that machine to hosts running Windows Server 2008 SP1 or later.

Example format: -DynamicMemoryEnabled \$True

Aliases	none
Required?	false
Position?	named

Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-DynamicMemoryMaximumMB<Int32>**

Specifies the maximum amount of memory that can be allocated to a virtual machine if dynamic memory is enabled. The default value is 65536.

REQUIRED: You can enable dynamic memory for a virtual machine only if that virtual machine is deployed on a host running Windows Server 2008 R2 SP1 or later or if the virtual machine is stored in a library in a stopped state (hardware changes to a stored virtual machine can only be made if the virtual machine does not have snapshots). Enabling dynamic memory on a virtual machine stored in a library will limit placement of that machine to hosts running Windows Server 2008 R2 SP1 or later.

Example format: -DynamicMemoryMaximumMB 1024

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-DynamicMemoryMinimumMB<UInt16>**

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-HardwareProfile<HardwareProfile>**

Specifies a hardware profile object.

---

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-HAVMPriority<UInt32>**

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-HighlyAvailable<Boolean>**

Specifies that a virtual machine will be placed on a Hyper-V host that is part of a host cluster. Configure this setting on a virtual machine, or on a template or hardware profile that will be used to create virtual machines.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-JobGroup<Guid>**

Specifies an identifier for a series of commands that will run as a set just before the final command that includes the same job group identifier runs.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-MemoryMB<Int32>**

Specifies, in megabytes (MB), the amount of random access memory (RAM) on the host that is allocated to a virtual machine. The default value is 512 MB. For a virtual machine on which dynamic memory is enabled (on a host running Windows Server 2008 R2 SP1 or later), use MemoryMB to specify the startup memory value.

TYPE OF HOST                    MAXIMUM HOST MEMORY ASSIGNABLE TO VM

-----                    -----

Hyper-V                    Up to 65536 MB RAM per virtual machine

VMware ESX Server 3.0.x Up to 16384 MB RAM per virtual machine

VMware ESX Server 3.5.x Up to 65532 MB RAM per virtual machine

Citrix XenServer    Up to 32265 MB RAM per VM

Example format: -MemoryMB 1024

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-MemoryPerVirtualNumaNodeMaximumMB<Int32>**

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-MemoryWeight<Int32>**

Indicates the priority in allocating memory to a virtual machine, relative to other virtual machines on the same host. A virtual machine with a higher setting is allocated more memory resources than a virtual machine with a lower setting.

For a host running Windows Server 2008 R2 SP1 or later, 5000 = Normal, 10000 = High, 0 = Low, 1 to 10000 = Custom.

NOTE: See the examples for a specific cmdlet to determine how that cmdlet uses this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-MonitorMaximumCount<Int32>**

Specifies the maximum number of monitors supported by a virtual video adapter.

Example format: -MonitorMaximumCount 3

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-MonitorMaximumResolution<String>**

Specifies, as a string, the value that represents the maximum possible monitor resolution of a virtual video adapter. Valid values are: "1024x768", "1280x1024", "1600x1200", "1920x1200". Default value: "1280x1024"

Example format: -MonitorResolutionMaximum "1600x1200"

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	false

Accept Wildcard Characters?	false
-----------------------------	-------

### **-NetworkUtilizationMbps<Int32>**

Specifies, in megabits per second (Mbps), the amount of bandwidth on the host's network that can be used by a specific virtual machine.

Example format: -NetworkUtilization 10

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-NumIsolationRequired<Boolean>**

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-NumLock<Boolean>**

Enables the BIOS value for NumLock on a virtual machine (or on a template or hardware profile that is used to create virtual machines) on a Hyper-V host. This parameter does not apply to virtual machines on VMware ESX hosts, or on Citrix XenServer hosts.

Aliases	none
Required?	false
Position?	named

Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Owner<String>**

Specifies the owner of a VMM object in the form of a valid domain user account.

Example format: -Owner "Contoso\ReneeLo"

Example format: -Owner "ReneeLo@Contoso"

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
---------	------



Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-UserRole<UserRole>**

Specifies a user role object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VirtualNumaNodesPerSocketMaximum<Byte>**

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VirtualVideoAdapterEnabled<Boolean>**

Enables, when set to \$True, the Microsoft Synthetic 3D Virtual Video Adapter for virtual machines. You can enable the Virtual Video Adapter directly on a virtual machine, or on a template or hardware profile that will be used to create virtual machines.

REQUIRED: You can enable the Microsoft Synthetic 3D Virtual Video Adapter for a virtual machine only if that virtual machine is deployed on a host running Windows Server 2008 R2 SP1 (with the Remote

---

Desktop Services role and Remote Desktop Virtual Graphics role service installed) or later or if the virtual machine is stored in a library in a stopped state (hardware changes to a stored virtual machine can only be made if the virtual machine does not have snapshots). Enabling the Microsoft Synthetic 3D Virtual Video Adapter on a virtual machine stored in a library will limit placement of that machine to hosts running Windows Server 2008 R2 SP1 (with the Remote Desktop Services role and Remote Desktop Virtual Graphics role service installed) or later.

Example format: -VirtualVideoAdapterEnabled \$TRUE

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **HardwareProfile**

---

## Examples

### 1: Create a default hardware profile.

This command creates a default hardware profile named NewHWProfile01.

```
PS C:\> New-SCHardwareProfile -Name "NewHWProfile01"
```

### 2: Create a hardware profile that sets boot order, CPU, and memory.

This command creates a new hardware profile, names it "NewHWProfile02", sets "PXEBoot" as the first entry in the BIOS boot order, specifies 1024 MB of memory, and specifies that a virtual machine created by using this hardware profile will have four processors.

```
PS C:\> New-SCHardwareProfile -Name "NewHWProfile02" -BootOrder  
PXEBoot,CD,Floppy,IDEHardDrive -MemoryMB 1024 -CPUCount 4
```

### 3: Clone and then modify an existing hardware profile.

The first command gets the hardware profile object named NewHWProfile01 and stores the object in the \$HWProfile variable.

The second command creates a hardware profile called NewHWProfile03 based on NewHWProfile01, but modifies the value for relative weight. All other settings in NewHWProfile03 are identical to those in NewHWProfile01.

```
PS C:\> $HWProfile = Get-SCHardwareProfile | where { $_.Name -eq "NewHWProfile01" }  
PS C:\> New-SCHardwareProfile -Name "NewHWProfile03" -HardwareProfile $HWProfile -  
RelativeWeight 100
```

### 4: Create a hardware profile that contains a network adapter, a SCSI adapter, and a DVD drive.

The first command generates a globally unique identifier (GUID) and stores the GUID string in variable \$JobGroupID. The job group ID functions as an identifier that groups subsequent commands that include this identifier into a single job group.

The second command creates a virtual network adapter but uses the JobGroup parameter to specify that the network adapter is not created until just before the New-SCHardwareProfile cmdlet in the last command runs. This command sets the physical (MAC) address type to dynamic and specifies that the new virtual network adapter will connect to a virtual network called "Internal Network."

The third command creates a virtual SCSI adapter but uses the JobGroup parameter to specify that the SCSI adapter is not created until just before the New-SCHardwareProfile cmdlet in the last command runs. This command sets the adapter ID to 6, and it sets the Shared parameter to \$False so that the adapter will not be shared (as it would have had to be if you wanted to use the adapter in guest clustering).

---

The fourth command creates a virtual DVD drive but uses the JobGroup parameter to specify that the DVD drive is not created until just before the New-SCHardwareProfile cmdlet in the last command runs. Specifying Bus 1 and LUN 0 attaches the virtual DVD drive to Secondary Channel (0) on the IDE bus. The last command creates a hardware profile named NewHWProfile04, sets the owner to Contoso\Katarina, specifies a description, and specifies that the amount of memory on the host that a virtual machine created by using this hardware profile will use is 512 MB. Before the New-SCHardwareProfile cmdlet creates the hardware profile, the JobGroup parameter in this final command executes all of the preceding cmdlets that specify the same JobGroup GUID. When New-SCVirtualNetworkAdapter, New-SCVirtualSCSIAdapter, and New-SCVirtualDVDDrive run, the resulting objects that are created are automatically associated with the new hardware profile.

```
PS C:\> $JobGroupId = [Guid]::NewGuid().ToString()
PS C:\> New-SCVirtualNetworkAdapter -JobGroup $JobGroupId -PhysicalAddressType Dynamic -
VirtualNetwork "Internal Network"
PS C:\> New-SCVirtualSCSIAdapter -JobGroup $JobGroupId -AdapterID 6 -Shared $False
PS C:\> New-SCVirtualDVDDrive -JobGroup $JobGroupId -Bus 1 -LUN 0
PS C:\> New-SCHardwareProfile -Name "NewHWProfile04" -Owner "Contoso\Katarina" -Description
"Temporary Hardware Config used to create a VM/Template" -MemoryMB 512 -JobGroup $JobGroupId
```

## 5: Create a hardware profile and add it to a new virtual machine template.

The first command creates a new hardware profile, names it "NewHWProfile05", specifies that it contains four processors and that the highest percentage of the total resources of a single CPU on a host that can be used by a virtual machine is 100 percent, assigns 64 GB of RAM and an owner, sets the HighlyAvailable flag \$True, and then stores the new hardware profile object in the \$HWProfile variable. The HighlyAvailable flag specifies that a virtual machine created by using this hardware profile, either directly or through a template, will be placed on a host that is a node of a host cluster.

The second command gets the virtual hard disk object named VHD01 from the library and stores the object in the \$VHD variable.

The third command gets an operating system object by name and stores the object in the \$OS variable.

The last command creates a new virtual machine template, names it "LargeVMTemplate", and specifies that it use the operating system, hardware profile, and virtual hard disk retrieved or created in the preceding commands, without any customization to the operating system.

```
PS C:\> $HWProfile = New-SCHardwareProfile -Name "NewHWProfile05" -CPUCount 4 -MemoryMB
64000 -CPUMax 100 -Owner "Contoso\Katarina" -HighlyAvailable $True
PS C:\> $VHD = Get-SCVirtualHardDisk | where { $_.Name -eq "VHD01.vhd" -and
$_.LibraryServer.Name -eq "LibServer01.Contoso.com" }
PS C:\> $OS = Get-SCOperatingSystem | where { $_.Name -eq "64-bit edition of Windows Server
2008 R2 Datacenter" }
PS C:\> New-SCVMTemplate -Name "LargeVMTemplate" -HardwareProfile $HWProfile -
OperatingSystem $OS -VirtualHardDisk $VHD -NoCustomization
```

---

## Related topics

[Get-SCHardwareProfile](#)

[Remove-SCHardwareProfile](#)

[Set-SCHardwareProfile](#)

---

# New-SCLoadBalancerConnectionPersistence

---

## New-SCLoadBalancerConnectionPersistence

Creates a load balancer connection persistence object that is used when you create a load balancer virtual IP.

### Syntax

Parameter Set: Default

```
New-SCLoadBalancerConnectionPersistence -Name <String> [-Value <String> ] [ <CommonParameters> ]
```

### Detailed Description

The New-SCLoadBalancerConnectionPersistence cmdlet creates a load balancer connection persistence object that is used when you create a load balancer virtual IP.

For information about creating a load balancer virtual IP, type: "Get-Help New-SCLoadBalancerVIP - detailed".

### Parameters

#### **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

#### **-Value<String>**

Specifies a string used to attribute an object or property.

---

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **LoadBalancerConnectionPersistence**

## Examples

### 1: Create a load balancer connection persistence object.

This command creates a load balancer connection persistence object named Source IP and stores the object in the \$LBConnectionPersistence variable.

```
PS C:\> $LBConnectionPersistence = New-SCLoadBalancerConnectionPersistence -Name "Source IP"
-Value "255.255.255.0 "
```

## Related topics

[New-SCLoadBalancerVIP](#)

---

# New-SCLoadBalancerHealthMonitor

---

## New-SCLoadBalancerHealthMonitor

Creates a load balancer health monitor object that is used when you create a load balancer virtual IP.

### Syntax

Parameter Set: Default

```
New-SCLoadBalancerHealthMonitor -IntervalSeconds <Int32> -ProtocolName <String> -  
TimeoutSeconds <Int32> [-Name <String> ] [-NumberOfRetries <Int32> ] [-Request <String> ] [-  
Response <String> ] [ <CommonParameters>]
```

### Detailed Description

The New-SCLoadBalancerHealthMonitor cmdlet creates a load balancer health monitor object that is used when you create a load balancer virtual IP.

For information about creating a load balancer virtual IP, type: "Get-Help New-SCLoadBalancerVIP - detailed".

### Parameters

#### -IntervalSeconds<Int32>

Specifies the amount of time, in seconds, that a health monitor waits between sending recurring requests to a load balancer to verify that the load balancer is available. The interval value should be larger than the timeout value.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false



---

## **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-NumberOfRetries<Int32>**

Specifies the number of times that a load balancer health monitor retries sending a request before marking the VIP member as down.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-ProtocolName<String>**

Specifies the protocol used to communicate with a load balancer.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-Request<String>**

Specifies the request that a health monitor sends to a load balancer. Typically, this command will make an HTTP GET request for the home page of the load balancer and check for a header response such as 200 OK.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Response<String>**

Specifies the expected response to a request that a health monitor sends to a load balancer.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-TimeoutSeconds<Int32>**

Specifies the amount of time, in seconds, that a process waits before timing out.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **LoadBalancerHealthMonitor**

## Examples

### 1: Create a load balancer health monitor.

This command creates a load balancer health monitor object named HTTPMonitor and stores the object in the \$LBHealthMonitor variable.

```
PS C:\> $LBHealthMonitor = New-SCLoadBalancerHealthMonitor -Name HTTPMonitor -ProtocolName  
"HTTP" -Request "GET /Index.html HTTP/1.1" -Response 200 -IntervalSeconds 15 -TimeoutSeconds  
20
```

## Related topics

[New-SCLoadBalancerVIP](#)

---

# New-SCLoadBalancerProtocol

---

## New-SCLoadBalancerProtocol

Creates a load balancer protocol object that is used when you create a load balancer virtual IP.

### Syntax

Parameter Set: Base

```
New-SCLoadBalancerProtocol -Name <String> [ <CommonParameters>]
```

Parameter Set: HTTPS

```
New-SCLoadBalancerProtocol -Name <String> [-HTTSCertificateSubjectName <String> ] [-  
HTTPSReencryptConnection <Boolean> ] [-TerminateHTTPS <Boolean> ] [ <CommonParameters>]
```

### Detailed Description

The New-SCLoadBalancerProtocol cmdlet creates a load balancer protocol object that is used when you create a load balancer virtual IP.

For information about creating a load balancer virtual IP, type: "Get-Help New-SCLoadBalancerVIP - detailed".

### Parameters

#### **-HTTSCertificateSubjectName<String>**

Specifies the subject name property of the certificate used to terminate the HTTPS connection at the load balancer.

Example format:

```
C=US,ST=WA,L=Redmond,O=Contoso,OU=Test,CN=www.contoso.com/emailAddress=contoso@contoso.com
```

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false

---

Accept Wildcard Characters?	false
-----------------------------	-------

### **-HTTPSReencryptConnection<Boolean>**

Indicates whether a load balancer should re-encrypt traffic to the server after it has terminated an HTTPS connection.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-TerminateHTTPS<Boolean>**

Indicates whether HTTPS traffic is terminated at the load balancer. If set to \$True, a certificate subject name must be provided.

Aliases	none
Required?	false
Position?	named

---

Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **LoadBalancerProtocol**

## Examples

### 1: Create an HTTPS load balancer protocol object.

This command creates a load balancer protocol object specifying that HTTPS terminates at the load balancer and that the load balancer re-encrypts the connection to the server. The command then stores the object in the \$LPProtocol variable.

```
PS C:\> $LBProtocol = New-SCLoadBalancerProtocol -Name HTTPS -HTTSCertificate  
"C=US,ST=WA,L=Redmond,O=Contoso,OU=Test,CN=www.contoso.com/emailAddress=contoso@contoso.com"  
-HTTPSReencryptconnection $True -TerminateHTTPS $True
```

## Related topics

[New-SCLoadBalancerVIP](#)

---

# New-SCLoadBalancerTemplate

---

## New-SCLoadBalancerTemplate

Creates a load balancer template that can be added to a service template.

### Syntax

Parameter Set: Default

```
New-SCLoadBalancerTemplate -ComputerTierTemplate <ComputerTierTemplate> -  
LoadBalancerVIPTemplate <LoadBalancerVIPTemplate> -VirtualNetworkAdapter  
<VirtualNetworkAdapter> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-  
VMNetworkServiceSetting <String> ] [-VMNetworkVIP <VMNetwork> ] [ <CommonParameters>]
```

Parameter Set: LogicalNetworkVIP

```
New-SCLoadBalancerTemplate -ComputerTierTemplate <ComputerTierTemplate> -  
LoadBalancerVIPTemplate <LoadBalancerVIPTemplate> -LogicalNetworkVIP <LogicalNetwork> -  
VirtualNetworkAdapter <VirtualNetworkAdapter> [-JobVariable <String> ] [-PROTipID <Guid> ]  
[-RunAsynchronously] [-VMNetworkServiceSetting <String> ] [-VMNetworkVIP <VMNetwork> ] [  
<CommonParameters>]
```

### Detailed Description

The New-SCLoadBalancerTemplate creates a load balancer template that you can add to a service template. When you deploy a service instance based on the service template, Virtual Machine Manager (VMM) will locate an appropriate load balancer in your VMM environment during placement, and configure it based on the properties provided in the load balancer template.

### Parameters

#### **-ComputerTierTemplate<ComputerTierTemplate>**

Specifies a computer tier template object.

Aliases	none
Required?	true
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-LoadBalancerVIPTemplate<LoadBalancerVIPTemplate>**

Specifies a load balancer virtual IP (VIP) template.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-LogicalNetworkVIP<LogicalNetwork>**

Specifies the logical networks from which the front-end IP address for the load balancer should be assigned (the front-end logical network affinity).

Aliases	none
Required?	true
Position?	named



Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VirtualNetworkAdapter<VirtualNetworkAdapter>**

Specifies a virtual network adapter object for a virtual machine.

TYPE OF HOST      NUMBER OF VIRTUAL NETWORK ADAPTERS

-----      -----

Hyper-V      Up to 4 emulated adapters per virtual machine.

Up to 8 synthetic adapters per virtual machine.

(Exception: no driver available for an emulated

---

network adapter on a Windows Server 2003 x64 guest.)

VMware ESX Up to 4 emulated adapters per virtual machine.

Citrix XenServer Up to 7 emulated adapters per virtual machine.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMNetworkServiceSetting<String>**

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMNetworkVIP<VMNetwork>**

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **LoadBalancerTemplate**

## Examples

### 1: Create a load balancer template.

The first command gets the service template object named ServiceTemplate01 and stores the object in the \$ServiceTemplate variable.

The second command gets the computer tier template for the service template stored in \$ServiceTemplate (in this case, ServiceTemplate01) and stores the object in the \$TierTemplate variable.

The third command gets the load balancer VIP template with the manufacturer of LB Manufacturer and model LB01, and stores the template in the \$LBVIPTemplate variable.

The fourth command gets the virtual machine template for the computer tier template stored in \$TierTemplate.

The fifth command gets the virtual network adapter for the virtual machine template stored in \$VMTemplate.

The sixth command creates a load balancer template using the computer tier template, load balancer template, and virtual network adapter objects obtained in the previous commands, and stores the object in the \$LBTemplate variable.

The last command displays information about the load balancer template to the user.

```
PS C:\> $ServiceTemplate = Get-SCServiceTemplate -Name "ServiceTemplate01"
PS C:\> $TierTemplate = Get-SCComputerTierTemplate -ServiceTemplate $ServiceTemplate
PS C:\> $LBVIPTemplate = Get-SCLoadBalancerVIPTemplate -Manufacturer "LB Manufacturer" -
Model "LB01"
PS C:\> $VMTemplate = Get-SCVMTemplate -ComputerTierTemplate $TierTemplate
PS C:\> $Adapter = Get-SCVirtualNetworkAdapter -VMTemplate $VMTemplate
PS C:\> $LBTemplate = New-SCLoadBalancerTemplate -ComputerTierTemplate $TierTemplate -
LoadBalancerVIPTemplate $LBVIPTemplate -VirtualNetworkAdapter $Adapter
PS C:\> $LBTemplate
```

## Related topics

[Get-SCLoadBalancerTemplate](#)

[Remove-SCLoadBalancerTemplate](#)

---

[Set-SCLoadBalancerTemplate](#)

---

# New-SCLoadBalancerVIP

---

## New-SCLoadBalancerVIP

Creates a load balancer VIP on a load balancer.

### Syntax

Parameter Set: Default

```
New-SCLoadBalancerVIP -IPAddress <String> -LoadBalancer <LoadBalancer> -Name <String> [-IsPersistenceEnabled <Boolean> ] [-JobVariable <String> ] [-LoadBalancerConnectionPersistence <LoadBalancerConnectionPersistence> ] [-LoadBalancerHealthMonitor <LoadBalancerHealthMonitor[]> ] [-LoadBalancerPort <UInt16> ] [-LoadBalancerProtocol <LoadBalancerProtocol> ] [-LoadBalancerVIPTemplate <LoadBalancerVIPTemplate> ] [-LoadBalancingMethod <LoadBalancingMethod> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The New-SCLoadBalancerVIP cmdlet creates a load balancer virtual IP (VIP) on a load balancer.

### Parameters

#### **-IPAddress<String>**

Specifies an IPv4 or IPv6 address.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

### **-IsPersistenceEnabled<Boolean>**

Indicates whether persistence is enabled for a load balancer virtual IP (VIP). When set to \$True, the load balancer will always attempt to direct a particular client to the same virtual machine behind the load balancer.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-LoadBalancer<LoadBalancer>**

Specifies a load balancer object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

-

### **-LoadBalancerConnectionPersistence<LoadBalancerConnectionPersistence>**

Specifies a load balancer connection persistence object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-LoadBalancerHealthMonitor<LoadBalancerHealthMonitor[]>**

Specifies a load balancer health monitor object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-LoadBalancerPort<UInt16>**

Specifies the port to use when configuring a virtual IP (VIP) in a load balancer.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-LoadBalancerProtocol<LoadBalancerProtocol>**

Specifies the protocol to use when connecting to a load balancer, or a load balancer protocol object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-LoadBalancerVIPTemplate<LoadBalancerVIPTemplate>**

Specifies a load balancer virtual IP (VIP) template.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-LoadBalancingMethod<LoadBalancingMethod>**

Specifies the load balancing method to use. Valid values are: RoundRobin, LeastConnectionsmember, Observedmember, Predictivemember, Ratiomember, Fastestmember, LeastConnections, Observednode, Predictivenode, Rationode, FastestResponseTime, LeastSessions, None

To determine the available methods for a specific load balancer, use the following command: (Get-SCLoadBalancer)[0].AvailableLoadBalancingMethods

Aliases	none
---------	------



---

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **LoadBalancerVIP**

## **Notes**

- Requires a VMM load balancer object, which can be retrieved using the Get-SCLoadBalancer cmdlet.

## **Examples**

### **1: Create a load balancer virtual IP (VIP).**

The first command creates a load balancer protocol object and stores the object in the \$LBProtocol variable.

---

The second command creates a load balancer connection persistence object and stores the object in the \$LBConnectionPersistence variable.

The third command creates a load balancer health monitor object and stores the object in the \$LBHealthMonitor variable.

The fourth command creates a load balancing method object and stores the object in the \$LBMethod variable.

The fifth command gets the load balancer object with the address LB01.Contoso.com and stores the object in the \$LoadBalancer variable.

The last command creates a load balancer VIP named LoadBalancerVIP01 using the load balancer stored in \$LoadBalancer and the objects created in the previous commands.

```
PS C:\> $LBProtocol = New-SCLoadBalancerProtocol -Name HTTPS -HTTPSCertificateSubjectName "C=US,ST=WA,L=Redmond,O=Contoso,OU=Test,CN=www.contoso.com/emailAddress=contoso@contoso.com" -HTTPSReencryptConnection $True -TerminateHTTPS $True
```

```
PS C:\> $LBConnectionPersistence = New-SCLoadBalancerConnectionPersistence -Name SourceIP -Value "255.255.255.0"
```

```
PS C:\> $LBHealthMonitor = New-SCLoadBalancerHealthMonitor -Name HTTPMonitor -Request "Get /Index.html HTTP/1.1" -Response 200 -IntervalSeconds 5 -TimeoutSeconds 2 -ProtocolName "HTTPS"
```

```
PS C:\> $LBMethod = New-SCLoadBalancingMethod -Name "LeastConnections"
```

```
PS C:\> $LoadBalancer = Get-SCLoadBalancer -LoadBalancerAddress "LB01.Contoso.com"
```

```
PS C:\> New-SCLoadBalancerVIP -Name "LoadBalancerVIP01" -IPAddress 10.0.0.1 -LoadBalancer $LoadBalancer -LoadBalancerConnectionPersistence $LBConnectionPersistence -LoadBalancerProtocol $LBProtocol -LoadBalancingMethod $LBMethod -LoadBalancerHealthMonitor $LBHealthMonitor -LoadBalancerPort 80
```

## Related topics

[Get-SCLoadBalancer](#)

[Get-SCLoadBalancerVIP](#)

[Read-SCLoadBalancerVIP](#)

[Remove-SCLoadBalancerVIP](#)

---

# New-SCLoadBalancerVIPMember

---

## New-SCLoadBalancerVIPMember

Adds a virtual machine to a load balancer VIP.

### Syntax

Parameter Set: Default

```
New-SCLoadBalancerVIPMember -IPAddress <String> -LoadBalancerVIP <LoadBalancerVIP> -Port <UInt16> -VirtualNetworkAdapter <VirtualNetworkAdapter> [-JobVariable <String> ] [-Name <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [<CommonParameters>]
```

### Detailed Description

The New-SCLoadBalancerVIPMember cmdlet adds a virtual machine to a load balancer virtual IP (VIP).

### Parameters

#### **-IPAddress<String>**

Specifies an IPv4 or IPv6 address.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

---

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-LoadBalancerVIP<LoadBalancerVIP>**

Specifies a virtual IP (VIP) in a load balancer.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Port<UInt16>**

Specifies the network port to use when adding an object or creating a connection. Valid values are: 1 to 4095.

---

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VirtualNetworkAdapter<VirtualNetworkAdapter>**

Specifies a virtual network adapter object for a virtual machine.

---

TYPE OF HOST      NUMBER OF VIRTUAL NETWORK ADAPTERS

-----      -----  
Hyper-V      Up to 4 emulated adapters per virtual machine.  
Up to 8 synthetic adapters per virtual machine.

(Exception: no driver available for an emulated network adapter on a Windows Server 2003 x64 guest.)

VMware ESX      Up to 4 emulated adapters per virtual machine.

Citrix XenServer Up to 7 emulated adapters per virtual machine.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

### **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **LoadBalancerVIPMember**

---

## Examples

### 1: Add a virtual machine to a load balancer virtual IP (VIP).

The first command gets the virtual machine object named VM01 on VMHost01 and stores the object in the \$VM variable.

The second command gets the virtual network adapter for the virtual machine stored in \$VM and stores the object in the \$VNIC variable.

The third command gets the load balancer with the address LB01.Contoso.com and stores the object in the \$LoadBalancer variable.

The fourth command gets the load balancer VIP with the address 10.0.0.1 for the load balancer stored in \$LoadBalancer and stores the object in the \$VIP variable.

The last command adds the virtual network adapter stored in \$VNIC (in this case, the virtual network adapter for VM01) to the load balancer VIP stored in \$VIP.

```
PS C:\> $VM = Get-VM -Name "VM01" -VMHost "VMHost01.Contoso.com"
```

```
PS C:\> $VNIC = Get-VirtualNetworkAdapter -VM $VM
```

```
PS C:\> $LoadBalancer = Get-SCLoadBalancer -LoadBalancerAddress "LB01.Contoso.com"
```

```
PS C:\> $VIP = Get-SCLoadBalancerVIP -LoadBalancer $LoadBalancer -IPAddress "10.0.0.1"
```

```
PS C:\> New-SCLoadBalancerVIPMember -LoadBalancerVIP $VIP -IPAddress "10.0.0.1" -Port 35 -  
VirtualNetworkAdapter $VNIC
```

## Related topics

[Disable-SCLoadBalancerVIPMember](#)

[Enable-SCLoadBalancerVIPMember](#)

[Get-SCLoadBalancer](#)

[Get-SCLoadBalancerVIP](#)

[Get-SCLoadBalancerVIPMember](#)

[Remove-SCLoadBalancerVIPMember](#)



---

# New-SCLoadBalancerVIPTemplate

---

## New-SCLoadBalancerVIPTemplate

Creates a load balancer VIP template used to create a load balancer VIP.

### Syntax

Parameter Set: Generic

```
New-SCLoadBalancerVIPTemplate -LoadBalancerPort <UInt16> -LoadBalancerProtocol <LoadBalancerProtocol> -LoadBalancingMethod <LoadBalancingMethod> -Name <String> [-Description <String> ] [-JobVariable <String> ] [-LoadBalancerConnectionPersistence <LoadBalancerConnectionPersistence> ] [-LoadBalancerHealthMonitor <LoadBalancerHealthMonitor[]> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: Specific

```
New-SCLoadBalancerVIPTemplate -LoadBalancerManufacturer <String> -LoadBalancerModel <String> -LoadBalancerPort <UInt16> -LoadBalancerProtocol <LoadBalancerProtocol> -LoadBalancingMethod <LoadBalancingMethod> -Name <String> [-Description <String> ] [-JobVariable <String> ] [-LoadBalancerConnectionPersistence <LoadBalancerConnectionPersistence> ] [-LoadBalancerHealthMonitor <LoadBalancerHealthMonitor[]> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The New-SCLoadBalancerVIPTemplate cmdlet creates a load balancer virtual IP (VIP) template used to create a load balancer VIP.

For information about creating a load balancer VIP, type: "Get-Help New-SCLoadBalancerVIP -detailed".

### Parameters

#### **-Description<String>**

States a description for the specified object.

Aliases	none
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

-

### **LoadBalancerConnectionPersistence<LoadBalancerConnectionPersistence>**

Specifies a load balancer connection persistence object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-LoadBalancerHealthMonitor<LoadBalancerHealthMonitor[]>**

Specifies a load balancer health monitor object.

---

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-LoadBalancerManufacturer<String>**

Specifies the name of the company that manufactured a load balancer.

Valid characters include: letters (a-z), numbers (0-9), underscore (\_), hyphen(-), dot(.), and single quote(').

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-LoadBalancerModel<String>**

Specifies the model of a load balancer

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-LoadBalancerPort<UInt16>**

Specifies the port to use when configuring a virtual IP (VIP) in a load balancer.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-LoadBalancerProtocol<LoadBalancerProtocol>**

Specifies the protocol to use when connecting to a load balancer, or a load balancer protocol object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-LoadBalancingMethod<LoadBalancingMethod>**

Specifies the load balancing method to use. Valid values are: RoundRobin, LeastConnectionsmember, Observedmember, Predictivemember, Ratiomember, Fastestmember, LeastConnections, Observednode, Predictivenode, Rationode, FastestResponseTime, LeastSessions, None

To determine the available methods for a specific load balancer, use the following command: (Get-SCLoadBalancer)[0].AvailableLoadBalancingMethods

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false

---

Accept Wildcard Characters?	false
-----------------------------	-------

### **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false

Accept Wildcard Characters?	false
-----------------------------	-------

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Examples**

### **1: Create a "specific" load balancer virtual IP (VIP) template.**

The first command creates a load balancer protocol object specifying that the HTTPS connection terminates at the load balancer, and then the connection is re-encrypted with the server. The command then stores the object in the \$LBProtocol variable.

The second command creates a load balancer connection persistence object with a value of 255.255.255.0, and then stores the object in the \$LBConnectionPersistence variable.

The third command creates a load balancer health monitor object specifying the load balancer protocol, the response, the interval in seconds, and the timeout in seconds. The command then stores the object in the \$LBHealthMonitor variable.

The fourth command creates a load balancer method object with the value of LeastConnections and stores the object in the \$LBMethod variable.

The last command creates a load balancer VIP template named VIPTemplate01 that is specific to the load balancer model LB01 manufactured by LB Manufacturer using the values for the objects stored in the \$LBConnectionPersistence, \$LBProtocol, \$LBMethod, and \$LBHealthMonitor created in the previous commands.

```
PS C:\> $LBProtocol = New-SCLoadBalancerProtocol -Name HTTPS -HTTPCertificate
"C=US,ST=WA,L=Redmond,O=Contoso,OU=Test,CN=www.contoso.com/emailAddress=contoso@contoso.com"
-HTTPSReencryptconnection $True -TerminateHTTPS $True
```

---

```
PS C:\> $LBConnectionPersistence = New-SCLoadBalancerConnectionPersistence -Name "Source IP"
-Value "255.255.255.0"
PS C:\> $LBHealthMonitor = New-SCLoadBalancerHealthMonitor -Name HTTPMonitor -ProtocolName
"HTTP" -Request "Get /Index.html HTTP/1.1" -Response 200 -IntervalSeconds 15 -TimeoutSeconds
20
PS C:\> $LBMethod = New-SCLoadBalancingMethod -Name "Least Connections"
PS C:\> New-SCLoadBalancerVIPTemplate -Name "VIPTemplate01" -Description "Specific virtual
IP Template" -LoadBalancerManufacturer "LB Manufacturer" -LoadBalancerModel "LB01" -
LoadBalancerPort "123" -LoadBalancerConnectionPersistence $LBConnectionPersistence -
LoadBalancerProtocol $LBProtocol -LoadBalancingMethod $LBMethod -LoadBalancerHealthMonitor
$LBHealthMonitor
```

## Related topics

[Get-SCLoadBalancerVIPTemplate](#)

[New-SCLoadBalancerVIP](#)

[Remove-SCLoadBalancerVIPTemplate](#)

[Set-SCLoadBalancerVIPTemplate](#)

---

# New-SCLoadBalancingMethod

---

## New-SCLoadBalancingMethod

Creates a load balancer method object that is used when you create a load balancer virtual IP.

### Syntax

Parameter Set: Default

```
New-SCLoadBalancingMethod -Name <String> [-Value <String> ] [ <CommonParameters>]
```

### Detailed Description

The New-SCLoadBalancingMethod cmdlet creates a load balancer method object that is used when you create a load balancer virtual IP.

For information about creating a load balancer virtual IP, type: "Get-Help New-SCLoadBalancerVIP -detailed".

### Parameters

#### **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

#### **-Value<String>**

Specifies a string used to attribute an object or property.



---

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **LoadBalancingMethod**

## Examples

### 1: Create a load balancing method object.

This command creates a load balancing method object named Least Connections and stores the object in the \$LBMethod variable.

```
PS C:\> $LBMethod = New-SCLoadBalancingMethod -Name "LeastConnections"
```

## Related topics

[New-SCLoadBalancerVIP](#)

---

# New-SCLogicalNetwork

---

## New-SCLogicalNetwork

Creates a logical network object.

### Syntax

Parameter Set: Default

```
New-SCLogicalNetwork [-Name] <String> [[-Description] <String> ] [-  
EnableNetworkVirtualization <Boolean> ] [-IsPVLAN <Boolean> ] [-JobVariable <String> ] [-  
LogicalNetworkDefinitionIsolation <Boolean> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-  
UseGRE <Boolean> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The New-SCLogicalNetwork cmdlet creates a Virtual Machine Manager (VMM) logical network object. Each logical network must have a unique name within a VMM installation.

Logical networks enable network administrators to model the network based on recognizable categories that align to business needs by grouping together subnets and VLANs. To assign IP subnets and VLANs to a logical network, create a logical network definition by using the New-SCLogicalNetworkDefinition cmdlet.

### Parameters

#### -Description<String>

States a description for the specified object.

Aliases	none
Required?	false
Position?	2
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

### **-EnableNetworkVirtualization<Boolean>**

Specifies whether network virtualization is enabled. The default value is \$False.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-IsPVLAN<Boolean>**

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-LogicalNetworkDefinitionIsolation<Boolean>**

Specifies whether logical network definition isolation is enabled.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-UseGRE<Boolean>**

Specifies whether network virtualization Generic Routing Encapsulation (GRE) is used. The default value is \$False.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **LogicalNetwork**

## Examples

### 1: Create a logical network.

This command creates a logical network named LogicalNetwork01.

```
PS C:\> New-SCLogicalNetwork -Name "LogicalNetwork01"
```

### 2: Create a network virtualization-enabled logical network.

This command creates a logical network named LogicalNetwork02 that is enabled for network virtualization and implements Generic Routing Encapsulation (GRE).

```
PS C:\> New-SCLogicalNetwork -Name "LogicalNetwork02" -EnableNetworkVirtualization $True -UseGRE $True
```

## Related topics

[Get-SCLogicalNetwork](#)

[New-SCLogicalNetworkDefinition](#)

[Remove-SCLogicalNetwork](#)

[Set-SCLogicalNetwork](#)

---

# New-SCLogicalNetworkDefinition

---

## New-SCLogicalNetworkDefinition

Creates a definition for a logical network that can be associated with one or more host groups.

### Syntax

Parameter Set: Default

```
New-SCLogicalNetworkDefinition -LogicalNetwork <LogicalNetwork> -Name <String> -SubnetVlan <SubnetVlan[]> -VMHostGroup <HostGroup[]> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The New-SCLogicalNetworkDefinition cmdlet creates a definition for a Virtual Machine Manager (VMM) logical network. The logical network can be associated with one or more host groups. A logical network definition is also called a network site.

After you create a new logical network, use the logical network definition to assign IP subnets and VLANs to the logical network. For information about creating logical networks, type: "Get-Help New-SCLogicalNetwork -detailed".

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-LogicalNetwork<LogicalNetwork>**

Specifies a logical network. A logical network is a named grouping of IP subnets and VLANs that is used to organize and simplify network assignments.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false



---

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-SubnetVlan<SubnetVlan[]>**

Specifies one or more IP subnet and VLAN sets.

For information about creating a SubnetVlan, type: "Get-Help New-SCSubNetVlan".

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMHostGroup<HostGroup[]>**

Specifies a virtual machine host group object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

### **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **LogicalNetworkDefiniton**

### **Notes**

- Requires a VMM logical network object, which can be retrieved by using the Get-SCLogicalNetwork cmdlet and a VMM host group object which can be retrieved by using the Get-SCVMHostGroup cmdlet.

### **Examples**

#### **1: Create a logical network definition for a logical network.**

The first command gets the logical network named LogicalNetwork01.

The second command creates a host group array and stores it in the \$HostGroup variable. The third and fourth commands retrieve the host groups named "HostGroup01" and "Production", and add them to the \$HostGroup array.

The fifth command creates a subnet VLAN array and stores it in the \$SubnetVLAN variable. The sixth and seventh commands create SubnetVLAN objects with the specified subnet and VLAN values and then store the objects in the \$SubnetVLAN array.

The eighth command creates a logical network definition named "Logical Network Definition 01" for the logical network object stored in the \$LogicanNetwork variable using the objects stored in the \$HostGroup and \$SubnetVLAN arrays.

---

```
PS C:\> $LogicalNetwork = Get-SCLogicalNetwork -Name "LogicalNetwork01"
PS C:\> $HostGroup =@()
PS C:\> $HostGroup += Get-SCVMHostGroup | where { $_.Path -eq "All Hosts\HostGroup01" }
PS C:\> $HostGroup += Get-SCVMHostGroup | where { $_.Path -eq "All
Hosts\HostGroup02\Production" }
PS C:\> $SubnetVLAN = @()
PS C:\> $SubnetVLAN += New-SCSubnetVLAN -Subnet 10.0.0.0/24 -VLAN 25
PS C:\> $SubnetVLAN += New-SCSubnetVLAN -Subnet FD4A:29CD:184F:3A2C::/64 -VLAN 25
PS C:\> New-SCLogicalNetworkDefinition -Name "Logical Network Definition 01" -LogicalNetwork
$LogicalNetwork -VMHostGroup $HostGroup -SubnetVLAN $SubnetVLAN
```

## Related topics

[New-SCSubnetVLAN](#)

[Get-SCLogicalNetworkDefinition](#)

[New-SCLogicalNetwork](#)

[Remove-SCLogicalNetworkDefinition](#)

[Set-SCLogicalNetworkDefinition](#)

---

# New-SCLogicalSwitch

---

## New-SCLogicalSwitch

Creates a logical switch.

### Syntax

Parameter Set: Default

```
New-SCLogicalSwitch -Name <String> [-Description <String> ] [-EnableSriov <Boolean> ] [-JobVariable <String> ] [-MinimumBandwidthMode {Default | Weight | Absolute | None} ] [-PROTipID <Guid> ] [-RunAsynchronously] [-SwitchUplinkMode {NoTeam | Team} ] [-VirtualSwitchExtensions <VirtualSwitchExtension[]> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The New-SCLogicalSwitch cmdlet creates a logical switch object.

### Parameters

#### **-Description<String>**

States a description for the specified object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

#### **-EnableSriov<Boolean>**

Specifies whether single-root I/O virtualization (SR-IOV) is enabled.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-MinimumBandwidthMode<BandwidthMode>**

Specifies the minimum bandwidth mode for the logical switch. Valid values are: Default, Weight, Absolute, None.

The acceptable values for this parameter are:

Default	
Weight	
Absolute	
None	

Aliases	none
Required?	false
Position?	named

---

Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false
Position?	named

Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-SwitchUplinkMode<SwitchUplinkMode>**

Specifies a switch uplink mode object.

The acceptable values for this parameter are:

NoTeam	
Team	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VirtualSwitchExtensions<VirtualSwitchExtension[]>**

Specifies an array of virtual switch extension objects.

To get a virtual switch extension object, use the Get-SCVirtualSwitchExtension cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

---

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Examples**

### **1: Create a logical switch.**

The first command gets the virtual switch extension object named VirtualSwitchExtension01 and stores the object in the \$VSwitchExt variable.

The second command creates a logical switch named LogicalSwitch01 using VirtualSwitchExtension01.

```
PS C:\> $VSwitchExt = Get-SCVirtualSwitchExtension -Name "VirtualSwitchExtension01"
```

```
PS C:\> New-SCLogicalSwitch -Name "LogicalSwitch01" -VirtualSwitchExtensions $VSwitchExt -  
MinimumBandwidthMode Default
```

## **Related topics**

[Get-SCLogicalSwitch](#)

[Remove-SCLogicalSwitch](#)

[Set-SCLogicalSwitch](#)



---

# New-SCMACAddressPool

---

## New-SCMACAddressPool

Creates a MAC address pool.

### Syntax

Parameter Set: Default

```
New-SCMACAddressPool -MACAddressRangeEnd <String> -MACAddressRangeStart <String> -Name <String> -VMHostGroup <HostGroup[]> [-Description <String> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters> ]
```

### Detailed Description

The New-SCMACAddressPool cmdlet creates a MAC address pool. A MAC address pool can be associated with one or more Virtual Machine Manager (VMM) host groups.

### Parameters

#### **-Description<String>**

States a description for the specified object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

---

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-MACAddressRangeEnd<String>**

Specifies the last address in a range of static MAC addresses.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-MACAddressRangeStart<String>**

Specifies the first address in a range of static MAC addresses.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Name<String>**

Specifies the name of a VMM object.

---

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMHostGroup<HostGroup[]>**

Specifies a virtual machine host group object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Notes**

- Requires a VMM host group object, which can be retrieved by using the Get-SCVMHostGroup cmdlet.

## **Examples**

### **1: Create a MAC address pool.**

The first command gets the host group named "Production" and stores it in the \$HostGroup variable.

The second command creates a MAC address pool named "MAC Address Pool 01" with an address range beginning with "00-1D-D8-B7-1C-00" and ending with "00-1D-D8-F4-1F-FF", as delineated by the MACAddressRangeStart and MACAddressRangeEnd parameters. The command also associates the pool with the host group stored in \$HostGroup.

---

```
PS C:\> $HostGroup = Get-SCVMHostGroup | where { $_.Path -eq "All  
Hosts\HostGroup02\Production" }
```

```
PS C:\> New-SCMACAddressPool -Name "MAC Address Pool 01" -VMHostGroup $HostGroup -  
MACAddressRangeStart "00-1D-D8-B7-1C-00" -MACAddressRangeEnd "00-1D-D8-F4-1F-FF"
```

## Related topics

[Get-SCMACAddressPool](#)

[Get-SCVMHostGroup](#)

[Remove-SCMACAddressPool](#)

[Set-SCMACAddressPool](#)

---

# New-SCNativeUplinkPortProfile

---

## New-SCNativeUplinkPortProfile

Creates a native uplink port profile.

### Syntax

Parameter Set: Default

```
New-SCNativeUplinkPortProfile [-Name] <String> [-Description <String> ] [-  
EnableNetworkVirtualization <Boolean> ] [-JobVariable <String> ] [-  
LBFOLoadBalancingAlgorithm {TransportPorts | IPAddresses | MacAddresses | HyperVPort} ] [-  
LBFOTeamMode {Static | SwitchIndependent | LACP} ] [-LogicalNetworkDefinition  
<LogicalNetworkDefinition[]> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer  
<ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The New-SCNativeUplinkPortProfile cmdlet creates a native uplink port profile object.

### Parameters

#### **-Description<String>**

States a description for the specified object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

#### **-EnableNetworkVirtualization<Boolean>**

Specifies whether network virtualization is enabled. The default value is \$False.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

-

### **LBFOLoadBalancingAlgorithm<NetworkAdapterLBFOLoadBalancingAlgorithm>**

The acceptable values for this parameter are:

TransportPorts	
IPAddresses	
MacAddresses	
HyperVPort	

Aliases	none
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-LBFOTeamMode<NetworkAdapterLBFOTeamMode>**

The acceptable values for this parameter are:

Static	
SwitchIndependent	
Lacp	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-LogicalNetworkDefinition<LogicalNetworkDefinition[]>**

Specifies a logical network definition (also called a network site) that contains the subnet that the IP address pool serves as specified by the Subnet parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false



---

## **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Examples**

### **1: Create a native uplink port profile by using a logical network definition.**

The first command gets the logical network definition object named Logical Network Definition 01 and stores the object in the \$LogNetDefinition variable.

The second command creates a native uplink port profile named NativeUplinkPortProfile01 by using Logical Network Definition 01, and enables network virtualization.

```
PS C:\> $LogNetDefinition = Get-SCLogicalNetworkDefinition -Name "Logical Network Definition 01"
```

```
PS C:\> New-SCNativeUplinkPortProfile -Name "NativeUplinkPortProfile01" -  
LogicalNetworkDefinition $LogNetDefinition -EnableNetworkVirtualization $True
```

## **Related topics**

[Get-SCNativeUplinkPortProfile](#)

[Remove-SCNativeUplinkPortProfile](#)

[Set-SCNativeUplinkPortProfile](#)

---

# New-SCOpsMgrConnection

---

## New-SCOpsMgrConnection

Creates a connection to an Operations Manager management group.

### Syntax

Parameter Set: Default

```
New-SCOpsMgrConnection -OpsMgrServer <String> -VMMServerCredential <PSCredential> [-EnableMaintenanceModeIntegration <Boolean> ] [-EnablePRO <Boolean> ] [-JobVariable <String> ] [-OpsMgrServerCredential <RunAsAccount> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-UseVMMServerServiceAccount] [-VMMServer <ServerConnection> ] [ <CommonParameters> ]
```

### Detailed Description

The New-SCOpsMgrConnection cmdlet creates a connection between the Virtual Machine Manager (VMM) management server that you are currently connected to and an Operations Manager management group.

Before creating a connection to an Operations Manager management server, you must install the Operations Manager management console on your VMM management server and install the following management packs in Operations Manager:

- SQL Server Core Library version 6.0.5000.0 or later management pack
- Windows Server Internet Information Services Library version 6.0.5000.0 or later management pack
- Windows Server Internet Information Services 2003 version 6.0.5000.0 or later management pack
- Windows Server Internet Information Services 7 version 6.0.6539.0 or later management pack

### Parameters

#### **-EnableMaintenanceModeIntegration<Boolean>**

Indicates whether maintenance mode integration is enabled for this connection.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-EnablePRO<Boolean>**

Indicates whether PRO is enabled for this connection.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-OpsMgrServer<String>**

Specifies the fully qualified domain name (FQDN) of the System Center Operations Manager management server to which VMM connects.

Aliases	none
Required?	true
Position?	named

Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-OpsMgrServerCredential<RunAsAccount>**

Specifies the credentials that VMM uses to connect to the Operations Manager management group. The specified account must be an administrator on the Operations Manager computer (a member of the Builtin\Administrators group).

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-UseVMMServiceAccount**

Specifies the service account that VMM uses to connect to Operations Manager.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-VMMServerCredential<PSCredential>**

Specifies the credentials that Operations Manager uses to connect with VMM.

Aliases	none
---------	------

---

Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **OpsMgrConnection**

## Examples

### 1: Create a connection to an Operations Manager management server.

The first command gets the Run As account object named OpsMgrServerAccount and stores the object in the \$OMCreds variable. The credentials provided in the Run As account must be a member of the Administrators user role on the Operations Manager management server.

The second command prompts you to for credentials, and stores the credentials you provide in the \$VMMCreds variable. The account you provide must be a member of the Administrator user role on the VMM management server.

The last command creates a connection between the VMM management server that you are currently connected to and OpsMgrServer01 using the credentials obtained in the first two commands.

```
PS C:\> $OMCreds = Get-SCRunAsAccount -Name "OpsMgrServerAccount"
```

```
PS C:\> $VMMCreds = Get-Credential
```

```
PS C:\> New-SCOpsMgrConnection -OpsMgrServer "OpsMgrServer01.Contoso.com" -  
OpsMgrServerCredential $OMCreds -VMMServerCredential $VMMCreds
```

## Related topics

[Get-SCOpsMgrConnection](#)

[Remove-SCOpsMgrConnection](#)

[Set-SCOpsMgrConnection](#)

[Write-SCOpsMgrConnection](#)

---

# New-SCP2V

---

## New-SCP2V

Converts a physical machine to a virtual machine on a Hyper-V host managed by VMM.

### Syntax

Parameter Set: \_\_AllParameterSets

```
New-SCP2V -VMHost <Host> [-AddDiskSizeMB <UInt64> ] [-Bus <Byte> ] [-CheckForUpdate] [-CPUCount <Byte> ] [-CPURelativeWeight <Int32> ] [-CPUType <ProcessorType> ] [-Credential <PSCredential> ] [-DelayStartSeconds <Int32> ] [-Description <String> ] [-DriverPath <String> ] [-Dynamic] [-EnableMACAddressSpoofing <Boolean> ] [-EnableVMNetworkOptimization <Boolean> ] [-Fixed] [-IDE] [-JobGroup <Guid> ] [-JobVariable <String> ] [-LogicalNetwork <LogicalNetwork> ] [-LUN <Byte> ] [-MACAddress <String> ] [-MACAddressType <String> ] [-MemoryMB <Int32> ] [-Name <String> ] [-NetworkLocation <String> ] [-NetworkTag <String> ] [-NoConnection] [-Offline] [-OfflineDefaultGateway <String> ] [-OfflineIPAddress <String> ] [-OfflineMACAddress <String> ] [-OfflinePrefixLength <String> ] [-OfflineSubnetMask <String> ] [-OverridePatchPath <String> ] [-Owner <String> ] [-Path <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-SCSI] [-Shutdown] [-SkipInstallVirtualizationGuestServices] [-SourceNetworkConnectionID <String> ] [-StartAction {NeverAutoTurnOnVM | AlwaysAutoTurnOnVM | TurnOnVMIfRunningWhenVSSStopped} ] [-StartVM] [-StopAction {SaveVM | TurnOffVM | ShutdownGuestOS} ] [-Trigger] [-UserRole <UserRole> ] [-VirtualNetwork <VirtualNetwork> ] [-VLANEnabled <Boolean> ] [-VLANID <UInt16> ] [-VMMServer <ServerConnection> ] [-VMNetwork <VMNetwork> ] [-VolumeDeviceID <String> ] [ <CommonParameters>]
```

Parameter Set: Install

```
New-SCP2V -SourceComputerName <String> -VMHost <Host> [-AddDiskSizeMB <UInt64> ] [-Bus <Byte> ] [-CheckForUpdate] [-CPUCount <Byte> ] [-CPURelativeWeight <Int32> ] [-CPUType <ProcessorType> ] [-Credential <PSCredential> ] [-Description <String> ] [-DriverPath <String> ] [-Dynamic] [-EnableMACAddressSpoofing <Boolean> ] [-EnableVMNetworkOptimization <Boolean> ] [-Fixed] [-IDE] [-JobGroup <Guid> ] [-JobVariable <String> ] [-LogicalNetwork <LogicalNetwork> ] [-LUN <Byte> ] [-MACAddress <String> ] [-MACAddressType <String> ] [-MemoryMB <Int32> ] [-Name <String> ] [-NetworkLocation <String> ] [-NetworkTag <String> ] [-NoConnection] [-Offline] [-OfflineDefaultGateway <String> ] [-OfflineIPAddress <String> ] [-OfflineMACAddress <String> ] [-OfflinePrefixLength <String> ] [-OfflineSubnetMask <String> ] [-OverridePatchPath <String> ] [-Owner <String> ] [-Path <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-SCSI] [-Shutdown] [-SkipInstallVirtualizationGuestServices] [-SourceNetworkConnectionID <String> ] [-StartVM] [-Trigger] [-UserRole <UserRole> ] [-VirtualNetwork <VirtualNetwork> ] [-VLANEnabled <Boolean> ] [-VLANID <UInt16> ] [-VMMServer <ServerConnection> ] [-VMNetwork <VMNetwork> ] [-VolumeDeviceID <String> ] [ <CommonParameters>]
```

Parameter Set: NoInstall

```
New-SCP2V -ComputerConfiguration <MachineConfiguration> -VMHost <Host> [-AddDiskSizeMB <UInt64> ] [-Bus <Byte> ] [-CheckForUpdate] [-CPUCount <Byte> ] [-CPURelativeWeight <Int32>
```



---

```

] [-CPUType <ProcessorType> ] [-Credential <PSCredential> ] [-Description <String> ] [-
DriverPath <String> ] [-Dynamic] [-EnableMACAddressSpoofing <Boolean> ] [-
EnableVMNetworkOptimization <Boolean> ] [-Fixed] [-IDE] [-JobGroup <Guid> ] [-JobVariable
<String> ] [-LogicalNetwork <LogicalNetwork> ] [-LUN <Byte> ] [-MACAddress <String> ] [-
MACAddressType <String> ] [-MemoryMB <Int32> ] [-Name <String> ] [-NetworkLocation <String>
] [-NetworkTag <String> ] [-NoConnection] [-Offline] [-OfflineDefaultGateway <String> ] [-
OfflineIPAddress <String> ] [-OfflineMACAddress <String> ] [-OfflinePrefixLength <String> ]
[-OfflineSubnetMask <String> ] [-OverridePatchPath <String> ] [-Owner <String> ] [-Path
<String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-SCSI] [-Shutdown] [-
SkipInstallVirtualizationGuestServices] [-SourceNetworkConnectionID <String> ] [-StartVM] [-
Trigger] [-UserRole <UserRole> ] [-VirtualNetwork <VirtualNetwork> ] [-VLANEnabled <Boolean>
] [-VLANID <UInt16> ] [-VMMServer <ServerConnection> ] [-VMNetwork <VMNetwork> ] [-
VolumeDeviceID <String> ] [ <CommonParameters>]

```

## Detailed Description

The New-SCP2V cmdlet converts a physical machine to a virtual machine on a Hyper-V host managed by Virtual Machine Manager (VMM). You cannot specify a VMware ESX host or a Citrix XenServer host as the destination host for the new virtual machine.

In a P2V conversion, you create a virtual machine from a physical source computer. The New-SCP2V cmdlet configures the new virtual machine to have the same hardware, software, and configuration settings as the physical source computer and to have the same identity (ComputerName.DomainName) as the source computer. During the P2V conversion, disk images of the hard disks on the physical computer are copied to Windows-based virtual hard disk files (.vhd files) for use in the new virtual machine.

For information about the operating systems for which VMM supports P2V, see "Converting Physical Computers to Virtual Machines in VMM" in the TechNet Library at <http://go.microsoft.com/fwlink/?LinkId=225101>.

### ADDING NEEDED FILES TO THE PATCH CACHE

-----

Some conversions might require that additional files be added to the internal cache. You can use the Add-SCPatch cmdlet to add the required files to the cache.

## Parameters

### **-AddDiskSizeMB<UInt64>**

Specifies, in megabytes (MB), the amount of additional disk space to add to a virtual hard disk when performing a physical-to-virtual (P2V) or virtual-to-virtual (V2V) machine conversion. Volumes located on the virtual hard disk are automatically extended to fill the entire virtual hard disk.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Bus<Byte>**

Specifies the IDE bus to which to attach a virtual disk drive or virtual DVD drive, or the SCSI bus to which to attach a virtual disk drive.

Example format: -IDE -Bus 1 -LUN 0

Example format: -SCSI -Bus 0 -LUN 1

NOTE: See the examples for a specific cmdlet to determine how that cmdlet uses this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-CheckForUpdate**

Checks the source computer for any patches required for conversion.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-ComputerConfiguration<MachineConfiguration>**

Specifies a computer configuration object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **-CPUCount<Byte>**

Specifies the number of CPUs on a virtual machine, on a hardware profile, or on a template. See the examples for a specific cmdlet to determine how that cmdlet uses this parameter.

TYPE OF HOST    NUMBER OF PROCESSORS

-----    -----

Hyper-V        Up to 4 CPUs per VM; varies by guest OS

VMware ESX    Up to 4 CPUs per VM for any supported guest OS

Exception: 1 CPU on a VM running Windows NT 4.0

Citrix XenServer Up to 8 CPUs per VM; varies by guest OS

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-CPURelativeWeight<Int32>**

Specifies the amount of CPU resources on a host that this virtual machine can use relative to other virtual machines on the same host. A virtual machine with a higher setting is allocated more CPU resources than a virtual machine with a lower setting.

TYPE OF HOST    RANGE OF RELATIVE VALUES

-----    -----

---

Hyper-V 1 to 10000

VMware ESX 2000 = High

1500 = Above Normal

1000 = Normal (default)

750 = Below Normal

500 = Low

1 to 1000000 = Custom

The VMware term for these values is "shares."

Citrix XenServer 1 to 65536, normal is 256

NOTE: See the examples for a specific cmdlet to determine how that cmdlet uses this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-CPUType<ProcessorType>**

Specifies the type of CPU for a virtual machine. To retrieve a list of all CPU types that are available for use in virtual machines in a VMM environment, type: "Get-SCCPUType"

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Credential<PSCredential>**

Specifies a credential object or, for some cmdlets, a Run As account object that contains the user name and password of an account that has permission to perform this action. Or, in the case of Restart-SCJob, has permission to complete a restarted task.

For more information about the PSCredential object, type: "Get-Help Get-Credential".

---

For more information about Run As accounts, type: "Get-Help New-SCRunAsAccount".

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-DelayStartSeconds<Int32>**

Specifies the number of seconds to wait after the virtualization service starts before automatically starting a virtual machine. This delay is used to stagger the startup time of multiple virtual machines to help reduce the demand on the physical computer's resources. A typical setting might be 30 to 60 seconds.

TYPE OF HOST      MAXIMUM CONFIGURABLE DELAY

-----

Hyper-V      1000000000 seconds (277777 hours)

VMware ESX      65535 seconds (18 hours)

Citrix XenServer      Does not apply to XenServer virtual machines

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Description<String>**

States a description for the specified object.

Aliases	none
Required?	false
Position?	named

---

Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-DriverPath<String>**

Specifies the path to drivers for an offline physical-to-virtual machine conversion (P2V conversion).

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Dynamic**

Specifies that a virtual hard disk can expand dynamically.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-EnableMACAddressSpoofing<Boolean>**

Enables, when set to \$True, MAC Address spoofing.

Aliases	none
Required?	false
Position?	named

Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-EnableVMNetworkOptimization<Boolean>**

Enables, when set to \$True, virtual machine network optimization. This feature improves network performance for virtual machines with network adapters that support virtual machine queue (VMQ) or TCP Chimney Offload. VMQ enables creating a unique network queue for each virtual network adapter. TCP Chimney Offload enables network traffic processing to be offloaded from the networking stack.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Fixed**

Specifies that a virtual hard disk is fixed in size.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-IDE**

Specifies IDE as the bus type to which to attach a virtual disk drive object or a virtual DVD drive object configured on a virtual machine or on a template.

Example format: -IDE -Bus 0 -LUN 1

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-JobGroup<Guid>**

Specifies an identifier for a series of commands that will run as a set just before the final command that includes the same job group identifier runs.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-LogicalNetwork<LogicalNetwork>**

Specifies a logical network. A logical network is a named grouping of IP subnets and VLANs that is used to organize and simplify network assignments.



Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-LUN<Byte>**

Specifies the logical unit number (LUN) for a virtual disk drive object or for a virtual DVD drive object on an IDE bus, or for a virtual disk drive object on a SCSI bus.

Example format: -IDE -Bus 1 -LUN 0

Example format: -SCSI -Bus 0 -LUN 1

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-MACAddress<String>**

Specifies the Media Access Control (MAC) address, or a set of MAC addresses, for a physical or virtual network adapter on a computer. Valid values are: one or more MAC addresses.

Example format for a single MAC address:

-MACAddress "00-15-5D-B4-DC-00"

Example format for a set of MAC addresses:

-MACAddress "00-15-5D-B4-DC-00", "00-1A-A0-E3-75-29"

Example format for a set of MAC addresses:

\$Macs = "00-15-5D-B4-DC-00", "00-1A-A0-E3-75-29"

Set-SCPXEServer -MACAddress \$Macs

NOTE: When used with New-SCPXEServer or Set-SCPXEServer, the MACAddress parameter updates the PXE interfaces from which the SCDM PXE Server listens for and responds to PXE requests

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-MACAddressType<String>**

Specifies the type of MAC address to use for a virtual network adapter. Valid values are: Static, Dynamic.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-MemoryMB<Int32>**

Specifies, in megabytes (MB), the amount of random access memory (RAM) on the host that is allocated to a virtual machine. The default value is 512 MB. For a virtual machine on which dynamic memory is enabled (on a host running Windows Server 2008 R2 SP1 or later), use MemoryMB to specify the startup memory value.

TYPE OF HOST                      MAXIMUM HOST MEMORY ASSIGNABLE TO VM

-----                      -----

Hyper-V                      Up to 65536 MB RAM per virtual machine

VMware ESX Server 3.0.x Up to 16384 MB RAM per virtual machine

VMware ESX Server 3.5.x Up to 65532 MB RAM per virtual machine

Citrix XenServer    Up to 32265 MB RAM per VM

Example format: -MemoryMB 1024

Aliases	none
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-NetworkLocation<String>**

Specifies the network location for a physical network adapter or for a virtual network adapter, or changes the default network location of a host's physical network adapter.

Example formats:

-NetworkLocation \$NetLoc (\$NetLoc might contain "Corp.Contoso.com")

-OverrideNetworkLocation \$TRUE -NetworkLocation "HostNICNewLocation.Contoso.com"

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-NetworkTag<String>**

Specifies a word or phrase to associate with a virtual network adapter that is configured to connect to a specific internal or external network on the host. The NetworkTag identifies all virtual machines with the same NetworkTag as members of the same network. VMM uses a NetworkTag (if one exists) when it evaluates hosts as possible candidates on which to deploy a virtual machine. If the host does not include virtual machines on the network with the same NetworkTag as the virtual machine to be placed, the host receives zero stars in the placement process.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-NoConnection**

Disconnects a virtual network adapter from a virtual network.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Offline**

Specifies that the operation is performed offline.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-OfflineDefaultGateway<String>**

Specifies the IP address of the gateway router that Windows PE uses during an offline P2V conversion. Valid formats: IPv4 or IPv6 address.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-OfflineIPAddress<String>**

Specifies an IPv4 or IPv6 address on the source computer that Windows PE uses during an offline P2V conversion.

**REQUIRED:** Use with the `OfflineSubnetMask` parameter (for an IPv4 address) or with the `OfflinePrefixLength` parameter (for an IPv4 or IPv6 address).

1. Use with the `SubnetMask` parameter for an IPv4 address.

Example IPv4 format: `-OfflineIPAddress 172.30.180.220 -OfflineSubnetMask 255.255.252.0`

2. Use with the `OfflinePrefixLength` parameter for an IPv4 or IPv6 address written in CIDR notation (IPAddress/PrefixLength).

Example IPv4 format: `-OfflineIPAddress 192.168.0.0 -OfflinePrefixLength 22`

The format above represents the IPv4 address typically written: `192.168.0.0/22`

This represents the following range of IPv4 addresses: `192.168.0.0 - 192.168.3.255`

Example IPv6 format: `-OfflineIPAddress 2001:DB8:: -OfflinePrefixLength 48`

The format above represents the IPv6 address typically written: `-IPAddress 2001:DB8::/48`

This represents the following range of IPv6 addresses: `2001:DB8:0:0:0:0:0 - 2001:DB8:0:FFFF:FFFF:FFFF:FFFF:FFFF`

Note: In an IPv6 address, you can use a double colon (::) to represent a series of zeros. However, the double colon can be used only once in an IPv6 address.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-OfflineMACAddress<String>**

Specifies the MAC address of the network interface card (NIC) on the source computer that Windows PE uses during an offline P2V conversion.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-OfflinePrefixLength<String>**

Specifies the length of the prefix for the IPv6 address on the source computer that Windows PE uses during an offline P2V conversion.

Example format: -OfflineIPAddress 2001:DB8:: -OfflinePrefixLength 48

REQUIRED: Use with the OfflineIPAddress parameter to specify IPv6 address.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-OfflineSubnetMask<String>**

Specifies the subnet mask. For example, use OfflineSubnetMask to specify the IPv4 address on the source computer that Windows PE uses during an offline P2V conversion.

Example format: -OfflineIPAddress "192.168.100.100" -OfflineSubnetMask "255.255.255.0"

REQUIRED: Use with the OfflineIPAddress parameter to specify an IPv4 address.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-OverridePatchPath<String>**

For internal use only (not for use in your code).

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Owner<String>**

Specifies the owner of a VMM object in the form of a valid domain user account.

Example format: -Owner "Contoso\ReneeLo"

Example format: -Owner "ReneeLo@Contoso"

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Path<String>**

Specifies the destination path for the operation.

Example formats:

Local path -Path "F:\"

UNC path -Path "\\Library\Templates"

Volume GUID path -Path "\\?\Volume{4703c1ea-8ae7-11db-b473-00123f7603e3}\"

VMware ESX path -Path "[storage1]\MyVMwareFolderForVMs\MyVM.vmx"

Citrix XenServer path - Path "Local storage[99b6212f-b63d-c676-25f9-d6c460992de7]"

Wildcards are supported for "Get" cmdlets and when you specify the UNC path:

Example format:

UNC path -Path "\\VMHostServer\MyVMs\*VM\*"

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false



---

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-SCSI**

Specifies SCSI as the bus type to which to attach a virtual disk drive object configured on a virtual machine or on a template.

Example format: -SCSI -Bus 0 -LUN 0

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Shutdown**

Indicates that a virtual machine, service, or a source server should be shut down. In the case of a virtual machine or service, the associated cmdlet attempts to use the operating system to shut the virtual machine down gracefully. In the case of a successful physical-to-virtual machine (P2V) conversion, the cmdlet shuts down the source server.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-SkipInstallVirtualizationGuestServices**

Skips the installation of virtualization guest services on a virtual machine. By default, this parameter is set to \$False and VMM installs the appropriate virtualization guest service automatically. For a virtual machine on a Hyper-V host, the virtualization guest service is called Integration Components (VMGuest.iso). For a virtual machine on a XenServer host, the virtualization guest service is called Citrix Tools for Virtual Machines (xs-tools.iso). Virtual machines on a VMware ESX host do not use a virtualization guest service.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-SourceComputerName<String>**

Specifies the source computer for a physical-to-virtual (P2V) machine conversion performed by VMM. Valid formats: FQDN, IPv4 or IPv6 address, or NetBIOS name.

Note: See the examples for a specific cmdlet to determine how that cmdlet specifies the source computer name.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-SourceNetworkConnectionID<String>**

Specifies the MAC address or network name of the physical network adapter to be converted into a virtual network adapter in the virtual machine.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-StartAction<VMStartAction>**

Specifies the behavior of a virtual machine when the virtualization service (Hyper-V, VMware, or XenServer) starts. Valid values are: AlwaysAutoTurnOnVM, NeverAutoTurnOnVM, TurnOnVMIfRunningWhenVSSStopped.

The acceptable values for this parameter are:

NeverAutoTurnOnVM	
AlwaysAutoTurnOnVM	
TurnOnVMIfRunningWhenVSSStopped	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-StartVM**

Specifies that the virtual machine starts when it arrives at the destination host.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-StopAction<VMStopAction>**

Specifies the behavior of the virtual machine when the virtualization service (Hyper-V, VMware, or XenServer) stops. Valid values are: SaveVM, TurnOffVM, ShutdownGuestOS.

The acceptable values for this parameter are:

SaveVM	
TurnOffVM	
ShutdownGuestOS	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Trigger**

Starts running the commands in a job group for a physical-to-virtual (P2V) conversion, a virtual-to-virtual (V2V) conversion, or the conversion of a physical hard disk to a virtual hard disk.

Aliases	none
Required?	false
Position?	named

Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-UserRole<UserRole>**

Specifies a user role object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VirtualNetwork<VirtualNetwork>**

Specifies a virtual network object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VlanEnabled<Boolean>**

Enables a virtual LAN (VLAN) for use by virtual machines on a Hyper-V or Citrix XenServer host.

Example format for a single VLAN: -VLANEnabled -VLANMode "Access" -VLANID 35

Example format for multiple VLANs: -VLANEnabled -VLANMode "Trunk" -VLANTrunkID 1,2,100,200,1124

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VlanID<UInt16>**

Assigns a numerical identifier in the range 1-4094 to a virtual network adapter on a virtual machine or to a physical network adapter on a virtual machine host.

Configure a VlanID on a Hyper-V, VMware ESX, or Citrix XenServer host:

- On an externally bound physical network adapter when the Vlan mode is Access.

Configure a VlanID on a virtual network adapter of a virtual machine:

- Bound to a physical network adapter on the host, or
- Bound to an internal virtual network on the host.

Example format: -VlanEnabled

-VlanMode "Access" -VLANID 35

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMHost<Host>**

Specifies a virtual machine host object. VMM supports Hyper-V hosts, VMware ESX hosts, and Citrix XenServer hosts.

For more information about each type of host, type: "Get-Help Add-SCVMHost -detailed". See the examples for a specific cmdlet to determine how that cmdlet uses this parameter.

Aliases	none
Required?	true

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-VMNetwork<VMNetwork>**

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VolumeDeviceID<String>**

Specifies the device ID of the volume to convert in a physical-to-virtual machine conversion (P2V conversion).

Aliases	none
Required?	false

---

Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **VirtualMachine**

## Examples

### 1: Convert a physical machine to a virtual machine.

The first command uses Get-Credential to prompt you to supply a user name and password and stores your credentials in the \$Credential variable. The required credentials for this operation are either a local Administrator account or a domain account with administrator rights on the computer that you want to convert.

The second command gets the host object named VMHost01 and stores the object in the \$VMHost variable.

The last command performs the following operations:

- Creates a virtual machine named P2VM01 from the source physical machine named P2VSource01 in the Contoso.com domain.
- Deploys the new virtual machine on the D: drive of VMHost01 in the VirtualMachinePath folder. In this example, all of the physical disks on P2VSource01.Contoso.com will be imaged and attached to the new virtual machine.
- Assigns 512 MB of memory on the host to the new virtual machine.
- Uses \$Credential to provide your credentials to New-SCP2V.
- Uses the -RunAsynchronously parameter to return control to the shell immediately (before the command completes).

```
PS C:\> $Credential = Get-Credential
```

```
PS C:\> $VMHost = Get-SCVMHost -ComputerName "VMHost01.Contoso.com"
```

```
PS C:\> New-SCP2V -SourceComputerName "P2VSource01.Contoso.com" -VMHost $VMHost -Name  
"P2VM01" -Path "D:\VirtualMachinePath" -MemoryMB 512 -Credential $Credential -  
RunAsynchronously
```



---

## Related topics

[Add-SCPatch](#)

[New-SCComputerConfiguration](#)

[Remove-SCComputerConfiguration](#)

---

# New-SCPackageMapping

---

## New-SCPackageMapping

Creates a package mapping object.

### Syntax

Parameter Set: Package

```
New-SCPackageMapping -TemplatePackage <Package> [-VMMServer <ServerConnection> ] [ <CommonParameters> ]
```

Parameter Set: Path

```
New-SCPackageMapping -Path <String> [-PreferPackageResources] [-VMMServer <ServerConnection> ] [ <CommonParameters> ]
```

### Detailed Description

The New-SCPackageMapping cmdlet creates a package mapping object. A package mapping object binds resources to a template. For information about how to update the bindings in a package mapping object, see Set-SCPackageMapping.

### Parameters

#### **-Path<String>**

Specifies the destination path for the operation.

Example formats:

Local path -Path "F:\"

UNC path -Path "\\Library\Templates"

Volume GUID path -Path "\\?\Volume{4703c1ea-8ae7-11db-b473-00123f7603e3}\"

VMware ESX path -Path "[storage1]\MyVMwareFolderForVMs\MyVM.vmx"

Citrix XenServer path - Path "Local storage[99b6212f-b63d-c676-25f9-d6c460992de7]"

Wildcards are supported for "Get" cmdlets and when you specify the UNC path:

Example format:

UNC path -Path "\\VMHostServer\MyVMs\\*VM\*"

Aliases	none
---------	------

Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-PreferPackageResources**

Indicates that the resources exported with the package are retained even if similar resources exist at the import destination.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-TemplatePackage<Package>**

Specifies an exported template package that contains seralized settings of a service or virtual machine template.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

---

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **PackageMapping**

## Examples

### 1: Create a package mapping for a template package.

The first command gets the template package at the specified path.

The second command creates a package mapping object for the package stored in \$TemplatePackage and stores the object in the \$Mappings variable.

```
PS C:\> $TemplatePackage = Get-SCTemplatePackage -Path "C:\TemplateExports\VMTemplate01.xml"
```

```
PS C:\> $Mappings = New-SCPackageMapping -TemplatePackage $TemplatePackage
```

## Related topics

[Set-SCPackageMapping](#)

---

# New-SCPortClassification

---

## New-SCPortClassification

Creates a port classification.

### Syntax

Parameter Set: Default

```
New-SCPortClassification [-Name] <String> [-Description <String> ] [-JobVariable <String> ]  
[-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [  
<CommonParameters>]
```

### Detailed Description

The New-SCPortClassification cmdlet creates a port classification.

### Parameters

#### **-Description<String>**

States a description for the specified object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## **Examples**

### **1. Create a port classification.**

This command creates a port classification named PortClass01.

```
PS C:\> New-SCPortClassification -Name "PortClass01"
```

## **Related topics**

[Get-SCPortClassification](#)

[Remove-SCPortClassification](#)

[Set-SCPortClassification](#)

---

# New-SCRunAsAccount

---

## New-SCRunAsAccount

Creates a VMM Run As account.

### Syntax

Parameter Set: Default

```
New-SCRunAsAccount [-Name] <String> -Credential <PSCredential> [-Description <String> ] [-JobGroup <Guid> ] [-JobVariable <String> ] [-NoValidation] [-Owner <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-UserRole <UserRole> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The New-SCRunAsAccount cmdlet creates a Run As account in Virtual Machine Manager (VMM).

### Parameters

#### **-Credential<PSCredential>**

Specifies a credential object or, for some cmdlets, a Run As account object that contains the user name and password of an account that has permission to perform this action. Or, in the case of Restart-SCJob, has permission to complete a restarted task.

For more information about the PSCredential object, type: "Get-Help Get-Credential".

For more information about Run As accounts, type: "Get-Help New-SCRunAsAccount".

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false



---

## **-Description<String>**

States a description for the specified object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-JobGroup<Guid>**

Specifies an identifier for a series of commands that will run as a set just before the final command that includes the same job group identifier runs.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-NoValidation**

Indicates that the Run As account will not validate the provided domain credentials.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Owner<String>**

Specifies the owner of a VMM object in the form of a valid domain user account.

Example format: -Owner "Contoso\ReneeLo"

Example format: -Owner "ReneeLo@Contoso"

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-UserRole<UserRole>**

Specifies a user role object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **RunAsAccount**

## **Examples**

### **1: Create a Run As account.**

The first command uses the Get-Credential cmdlet to prompt you to supply a user name and password and stores the credentials in the \$Credential variable.

The second command creates a Run As account named RunAsAccount01 using the credentials provided in \$Creds.

```
PS C:\> $Creds = Get-Credential
```

```
PS C:\> $RunAsAccount = New-SCRunAsAccount -Name "RunAsAccount01" -Credential $Creds
```

## **Related topics**

[Get-SCRunAsAccount](#)

[Remove-SCRunAsAccount](#)

[Set-SCRunAsAccount](#)

---

# New-SCScriptCommandSetting

---

## New-SCScriptCommandSetting

Creates a settings object for a script command.

### Syntax

Parameter Set: DefaultParamSet

```
New-SCScriptCommandSetting [-AlwaysReboot <Boolean> ] [-CommandMayReboot] [-FailOnMatch] [-MatchExitCode <String> ] [-MatchRebootExitCode <String> ] [-MatchStandardError <String> ] [-MatchStandardOutput <String> ] [-PersistStandardErrorPath <String> ] [-PersistStandardOutputPath <String> ] [-RestartOnRetry <Boolean> ] [-RestartScriptOnExitCodeReboot <Boolean> ] [-WarnAndContinueOnMatch] [-WorkingDirectory <String> ] [ <CommonParameters>]
```

### Detailed Description

The New-SCScriptCommandSetting cmdlet creates a settings object for a script command.

### Parameters

#### **-AlwaysReboot<Boolean>**

Indicates whether a computer or virtual machine should always restart after the script has finished running.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-CommandMayReboot**

Indicates that the script command may reboot the computer or virtual machine.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-FailOnMatch**

Indicates that the action taken when a failure policy is matched is to fail.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-MatchExitCode<String>**

Specifies the failure policy exit code.

Example format: `-MatchExitCode "[1-9][0-9]*"`

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-MatchRebootExitCode<String>**

Specifies the restart policy match exit code.

Example format: `-MatchRebootExitCode "{1641}{{3010}}{{3011}}"`

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-MatchStandardError<String>**

Specifies the failure policy standard error.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-MatchStandardOutput<String>**

Specifies the failure policy standard output.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-PersistStandardErrorPath<String>**

Specifies the file path to store the standard error.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-PersistStandardOutputPath<String>**

Specifies the file path to store the standard output.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-RestartOnRetry<Boolean>**

Specifies that a script is restarted upon VMM job restart if the previous job failure was due to a script failure when used in conjunction with New-ScriptCommandSetting or Set-SCScriptCommandSetting.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false



---

## **-RestartScriptOnExitCodeReboot<Boolean>**

Indicates whether the script restarts after the computer or virtual machine is restarted when an exit code is matched.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-WarnAndContinueOnMatch**

Indicates that the action taken when a failure policy is matched is to warn the user and continue the operation.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-WorkingDirectory<String>**

Specifies a working directory for a script command.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **ScriptCommandSetting**

## Examples

### 1: Add a working directory setting to a script command.

The first command gets the application profile object named SvcWebAppProfile01 and stores the object in the \$AppProfile variable.

The second command gets the script command object named PostInstall for the application profile stored in \$AppProfile.

The third command creates a new script command setting which sets the working directory to Working\_Folder\_02, and then stores the object in the \$ScriptSetting variable.

The last command updates the working directory for the script command stored in \$ScriptCommand to be Working\_Folder\_02 (the value of \$ScriptSetting).

```
PS C:\> $AppProfile = Get-SCApplicationProfile -Name "SvcWebAppProfile01"
PS C:\> $ScriptCommand = Get-SCScriptCommand -ApplicationProfile $AppProfile | where
{$_ .Name -eq "PostInstall"}
PS C:\> $ScriptSetting = New-SCScriptCommandSetting -WorkingDirectory "Working_Folder_02"
PS C:\> Set-SCScriptCommand -ScriptCommand $ScriptCommand -ScriptCommandSetting
$ScriptSetting
```

## Related topics

[Get-SCScriptCommandSetting](#)

[Set-SCScriptCommandSetting](#)

---

# New-SCService

---

## New-SCService

Deploys a new service instance in VMM.

### Syntax

Parameter Set: Default

```
New-SCService -ServiceConfiguration <ServiceConfiguration> [-JobVariable <String> ] [-Owner <String> ] [-PersistServiceConfiguration] [-PROTipID <Guid> ] [-RunAsynchronously] [-UserRole <UserRole> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The New-SCService cmdlet deploys a new service instance into the Virtual Machine Manager (VMM) environment. You can create a service directly from a service template if no service instance configuration is needed, or from a service configuration.

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

#### **-Owner<String>**

Specifies the owner of a VMM object in the form of a valid domain user account.

Example format: -Owner "Contoso\ReneeLo"

---

Example format: -Owner "ReneeLo@Contoso"

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-PersistServiceConfiguration**

Indicates that the service deployment configuration is stored in the VMM library after the service is deployed.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-ServiceConfiguration<ServiceConfiguration>**

Specifies a service configuration object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **-UserRole<UserRole>**

Specifies a user role object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

### **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **Service**

### **Examples**

#### **1: Deploy a service template using a service configuration that is stored in the library.**

The first command gets the service configuration object named Contoso Service Configuration 01 and stores the object in the \$SvcConfig variable.

The second command runs placement to update the service configuration stored in \$SvcConfig.

The third command deploys the new service using the service configuration stored in \$SvcConfig.

The last command displays the properties of the newly deployed service to the user.

```
PS C:\> $SvcConfig = Get-SCServiceConfiguration -Name "Service01"
```

```
PS C:\> Update-SCServiceConfiguration -ServiceConfiguration $SvcConfig
```

```
PS C:\> $NewService = New-SCService -ServiceConfiguration $SvcConfig
```

```
PS C:\> $NewService
```

### **Related topics**

[Get-SCService](#)

---

[Read-SCService](#)

[Remove-SCService](#)

[Resume-SCService](#)

[Set-SCService](#)

[Start-SCService](#)

[Stop-SCService](#)

[Suspend-SCService](#)

[Update-SCService](#)

---

# New-SCServiceConfiguration

---

## New-SCServiceConfiguration

Creates a service configuration from a service template.

### Syntax

Parameter Set: Cloud

```
New-SCServiceConfiguration [-Name] <String> -Cloud <Cloud> -ServiceTemplate <ServiceTemplate> [-CostCenter <String> ] [-Description <String> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-ServicePriority <String> ] [-Tag <String> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: HostGroup

```
New-SCServiceConfiguration [-Name] <String> -ServiceTemplate <ServiceTemplate> -VMHostGroup <HostGroup> [-CostCenter <String> ] [-Description <String> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-ServicePriority <String> ] [-Tag <String> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The New-SCServiceConfiguration cmdlet creates a service configuration from a service template. The service configuration contains instance-specific values that are used when the service is deployed.

### Parameters

#### -Cloud<Cloud>

Specifies a private cloud object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false



---

### **-CostCenter<String>**

Specifies the cost center for a virtual machine so that you can collect data about the allocation of virtual machines (or resources allocated to virtual machines) to make use of in your billing system.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Description<String>**

States a description for the specified object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-ServicePriority<String>**

Specifies the priority for a service. Valid values are: Normal, Low, High. Default value: Normal.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-ServiceTemplate<ServiceTemplate>**

Specifies a service template object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **-Tag<String>**

Specifies a word or phrase to associate with an object so that you can search for all objects with the specified set of tags. You can search for a subset of tags, or you can search for the full set of tags.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-VMHostGroup<HostGroup>**

Specifies a virtual machine host group object or an array of host group objects.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **ServiceConfiguration**

---

## Examples

### 1: Create a service configuration for a host group.

The first command gets the host group object with the path All Hosts\HostGroup02\Production and stores the object in the `$HostGroup` variable.

The second command gets the service template object named `ServiceTemplate01` and stores it in the `$ServiceTemplate` variable.

The third command creates a service configuration object in the library for the host group stored in `$HostGroup` using the service template object stored in `$ServiceTemplate`. The command then stores the new service configuration object in the `$ServiceConfig` variable.

The last command displays the properties of the new service configuration object to the user.

```
PS C:\> $HostGroup = Get-SCVMHostGroup | where { $_.Path -eq "All
Hosts\HostGroup02\Production" }
PS C:\> $ServiceTemplate = Get-SCServiceTemplate -Name "ServiceTemplate01"
PS C:\> $ServiceConfig = New-SCServiceConfiguration -ServiceTemplate $ServiceTemplate -Name
"Service01" -VMHostGroup $HostGroup -Description "Contoso Service 01" -ServicePriority High
-CostCenter 1033
PS C:\> $ServiceConfig
```

### 2: Create a service configuration for a private cloud.

The first command gets the private cloud object named `Production` and stores the object in the `$Cloud` variable.

The second command gets the service template object named `ServiceTemplate01` and stores it in the `$ServiceTemplate` variable.

The third command creates a service configuration object in the library for the private cloud stored in `$Cloud` using the service template object stored in `$ServiceTemplate`. The command then stores the new service configuration object in the `$ServiceConfig` variable.

The last command displays the properties of the new service configuration object to the user.

```
PS C:\> $Cloud = Get-SCCloud -Name "Production"
PS C:\> $ServiceTemplate = Get-SCServiceTemplate -Name "ServiceTemplate01"
PS C:\> $ServiceConfig = New-SCServiceConfiguration -ServiceTemplate $ServiceTemplate -Name
"Service02" -Cloud $Cloud -Description "Contoso Cloud Service" -ServicePriority High -
CostCenter 1033
PS C:\> $ServiceConfig
```

## Related topics

[Get-SCServiceConfiguration](#)

[Get-SCServiceTemplate](#)

[Remove-SCServiceConfiguration](#)

---

[Set-SCServiceConfiguration](#)

[Update-SCServiceConfiguration](#)

---

# New-SCServiceTemplate

---

## New-SCServiceTemplate

Creates a service template used to create a service in VMM.

### Syntax

Parameter Set: Default

```
New-SCServiceTemplate [-Name] <String> -Release <String> [-Description <String> ] [-JobVariable <String> ] [-Owner <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-ServicePriority {Normal | Low | High} ] [-ServiceTemplate <ServiceTemplate> ] [-UseAsDefaultRelease <Boolean> ] [-UserRole <UserRole> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The New-SCServiceTemplate cmdlet creates a service template used to create a service in Virtual Machine Manager (VMM). A service template is a description of a service that contains a set of service templates which describe how the service should be deployed, configured, and serviced. Service templates are stored in the VMM library.

### Parameters

#### -Description<String>

States a description for the specified object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Owner<String>**

Specifies the owner of a VMM object in the form of a valid domain user account.

Example format: -Owner "Contoso\ReneeLo"

Example format: -Owner "ReneeLo@Contoso"

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false



---

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Release<String>**

Specifies a string that describes the release of a library resource. VMM automatically creates a release value for every resource imported into the library. After the resource has been imported, the string can be customized.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false

Accept Wildcard Characters?	false
-----------------------------	-------

### **-ServicePriority<ServicePriority>**

Specifies the priority for a service. Valid values are: Normal, Low, High. Default value: Normal.

The acceptable values for this parameter are:

Normal	
Low	
High	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-ServiceTemplate<ServiceTemplate>**

Specifies a service template object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-UseAsDefaultRelease<Boolean>**

Specifies that this release is used as the default release for the service template.

---

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-UserRole<UserRole>**

Specifies a user role object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **ServiceTemplate**

## Examples

### 1: Create a service template.

The first command creates a service template object named ServiceTemplate01 and stores the object in the \$SvcTemplate variable.

The second command displays information about the service template object to the user.

```
PS C:\> $SvcTemplate = New-SCServiceTemplate -Name "ServiceTemplate01" -Release "Beta" -  
Description "Service Template 01" -Owner "Contoso\Katarina"
```

```
PS C:\> $SvcTemplate
```

### 2: Clone a service template.

The first command gets the service template object named ServiceTemplate01 with a release of Beta and stores the object in the \$SvcTemplate variable.

The second command creates a clone of ServiceTemplate01 and gives it a release value of v1. The command then stores the service template object in the \$NewSvcTemplate variable.

The last command displays information about the cloned service template object to the user.

```
PS C:\> $SvcTemplate = Get-SCServiceTemplate -Name "ServiceTemplate01" | where { $_.Release  
-eq "Beta" }
```

```
PS C:\> $NewSvcTemplate = New-SCServiceTemplate -Name "ServiceTemplate01" -Release "v1" -  
ServiceTemplate $SvcTemplate
```

```
PS C:\> $NewSvcTemplate
```

## Related topics

[Get-SCServiceTemplate](#)

[Read-SCServiceTemplate](#)

[Remove-SCServiceTemplate](#)

[Resolve-SCServiceTemplate](#)

[Set-SCServiceTemplate](#)

[Test-SCServiceTemplate](#)

---

# New-SCServicingWindow

---

## New-SCServicingWindow

Creates a servicing window and the schedule for the servicing window.

### Syntax

Parameter Set: DailyFrequency

```
New-SCServicingWindow [-Name] <String> -DaysToRecur <Int32> [-Category <String> ] [-Description <String> ] [-JobVariable <String> ] [-MinutesDuration <Int32> ] [-Owner <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-StartDate <DateTime> ] [-StartTimeOfDay <DateTime> ] [-TimeZone <Int32> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: MonthlyFrequency

```
New-SCServicingWindow [-Name] <String> -DayOfMonth {First | DayTwo | DayThree | DayFour | DayFive | DaySix | DaySeven | DayEight | DayNine | DayTen | DayEleven | DayTwelve | DayThirteen | DayFourteen | DayFifteen | DaySixteen | DaySeventeen | DayEighteen | DayNineteen | DayTwenty | DayTwentyOne | DayTwentyTwo | DayTwentyThree | DayTwentyFour | DayTwentyFive | DayTwentySix | DayTwentySeven | DayTwentyEight | DayTwentyNine | DayThirty | DayThirtyOne | Last} [-Category <String> ] [-Description <String> ] [-JobVariable <String> ] [-MinutesDuration <Int32> ] [-MonthsToRecur <Int32> ] [-Owner <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-StartDate <DateTime> ] [-StartTimeOfDay <DateTime> ] [-TimeZone <Int32> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: MonthlyRelativeFrequency

```
New-SCServicingWindow [-Name] <String> -MonthlyScheduleDayOfWeek {Sunday | Monday | Tuesday | Wednesday | Thursday | Friday | Saturday} -WeekOfMonth {First | Second | Third | Fourth | Last} [-Category <String> ] [-Description <String> ] [-JobVariable <String> ] [-MinutesDuration <Int32> ] [-MonthsToRecur <Int32> ] [-Owner <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-StartDate <DateTime> ] [-StartTimeOfDay <DateTime> ] [-TimeZone <Int32> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: WeeklyFrequency

```
New-SCServicingWindow [-Name] <String> -WeeklyScheduleDayOfWeek <String> [-Category <String> ] [-Description <String> ] [-JobVariable <String> ] [-MinutesDuration <Int32> ] [-Owner <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-StartDate <DateTime> ] [-StartTimeOfDay <DateTime> ] [-TimeZone <Int32> ] [-VMMServer <ServerConnection> ] [-WeeksToRecur <Int32> ] [ <CommonParameters>]
```

---

## Detailed Description

The New-SCServicingWindow cmdlet creates a servicing window and the schedule for the servicing window. A servicing window is a scheduled timeframe during which maintenance work can be done on a virtual machine, a host, or a service.

## Parameters

### **-Category<String>**

Specifies a category for a servicing window.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-DayOfMonth<DayOfMonthType>**

Specifies the ordinal day of the month on which the schedule starts. For example, 4 indicates the fourth day of the month. "Last" indicates the last day of the month.

The default value is the integer that corresponds to the same day as specified in the StartDate parameter.

Valid integer values: 1 through 31.

Valid string values: "First", "Last"

The acceptable values for this parameter are:

First	
DayTwo	
DayThree	
DayFour	
DayFive	
DaySix	
DaySeven	
DayEight	

DayNine	
DayTen	
DayEleven	
DayTwelve	
DayThirteen	
DayFourteen	
DayFifteen	
DaySixteen	
DaySeventeen	
DayEighteen	
DayNineteen	
DayTwenty	
DayTwentyOne	
DayTwentyTwo	
DayTwentyThree	
DayTwentyFour	
DayTwentyFive	
DayTwentySix	
DayTwentySeven	
DayTwentyEight	
DayTwentyNine	
DayThirty	
DayThirtyOne	
Last	

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

### **-DaysToRecur<Int32>**

Specifies, in days, the amount of time between scheduled jobs.

Minimum value: 1

Maximum value: 999

Default value: 1

Example format (to schedule a job to recur every third day): -DaysToRecur 3

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Description<String>**

States a description for the specified object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
Required?	false
Position?	named



Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-MinutesDuration<Int32>**

Specifies, in minutes, a period of time. For example, when used with New-SCServicingWindow or Set-SCServicingWindow, use this parameter to specify the amount of time for which to put a server or service into maintenance mode.

Example format: -DurationMinutes 120

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-MonthlyScheduleDayOfWeek<DayOfWeek>**

Specifies the day of the week to run a job that occurs on a monthly schedule. You can specify only one day of the week. The default value is the current day (if today is Tuesday, Tuesday is the default). Valid values to specify a specific day are: Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday.

Requirement: Use with the parameter WeekOfMonth.

The acceptable values for this parameter are:

Sunday	
Monday	
Tuesday	
Wednesday	
Thursday	
Friday	
Saturday	

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-MonthsToRecur<Int32>**

Specifies, in months, the amount of time between scheduled service windows

Minimum value: 1

Maximum value: None

Default value: 1

Example format (to schedule a service window to recur every other month): -MonthsToRecur 2

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

---

## **-Owner<String>**

Specifies the owner of a VMM object in the form of a valid domain user account.

Example format: -Owner "Contoso\ReneeLo"

Example format: -Owner "ReneeLo@Contoso"

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-StartDate<DateTime>**

Specifies the date to start a service window. The default value is the current date. You can type a new date in the short date format for your locale. Or, you can pass a Date-Time object from Get-Date.

Example format that specifies a DateTime object:

\$StartDate = Get-Date -Year 2012 -Day 5 -Month 4

-StartDate \$StartDate

Example formats that specifies a string:

-StartDate "August 19, 2011"

-StartDate "8/19/2011"

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-StartTimeOfDay<DateTime>**

Specifies the time of day, or a time-span during a 24-hour period, to start a job or other operation. The default value is the current time.

Example format: -StartTimeOfDay "08:00"

Example format for 2 hrs from 6:00pm: -StartTimeOfDay "18:00-20:00"

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-TimeZone<Int32>**

Specifies a number (an index) that identifies a geographical region that shares the same standard time. For a list of time zone indexes, see "Microsoft Time Zone Index Values" at: <http://go.microsoft.com/fwlink/?LinkId=120935>. If no time zone is specified, the default time zone used for a virtual machine is the same time zone setting that is on the virtual machine host.

Example format to specify the GMT Standard Time zone: -TimeZone 085

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **-WeeklyScheduleDayOfWeek<String>**

Specifies one or more days of the week to run a job. The default value is the current day of the week. Valid values to specify an individual day by using a string: Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday. Valid values to specify a set of days in a week: Any set of two or more days separated by commas. Valid values to specify an individual day by using an integer: 1, 2, 3, 4, 5, 6, 7

Example format: -WeeklyScheduleDayOfWeek "Monday"

Example format: -WeeklyScheduleDayOfWeek "Monday, Wednesday, Friday"

Example format: -WeeklyScheduleDayOfWeek 5 (to specify a Friday)

Requirement: Use with StartTimeOfDay.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-WeekOfMonth<WeekOfMonthType>**

Specifies a week relative to the first day of the month, such as first, second, third, fourth, or last.

The acceptable values for this parameter are:

First	
Second	
Third	
Fourth	
Last	

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-WeeksToRecur<Int32>**

Specifies, in weeks, the amount of time between scheduled jobs.

Minimum value: 1

Maximum value: None

Default value: 1

Example format (to schedule a job to recur every other week): -WeeksToRecur 2

---

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **ServicingWindow**

## Examples

### 1: Schedule a daily servicing window.

The first command gets the current date and adds two days, then stores the result in the \$Date variable.

The second command creates a servicing window named Backup Staging A that occurs every third day at 1:30 PM in the GMT Standard time zone. The start date of the servicing window is set to the date stored in \$Date, which is in two days.

```
PS C:\> $Date = (Get-Date).AddDays(2)
```

```
PS C:\> New-SCServicingWindow -Name "Backup Staging A" -Category "Non Essential" -StartDate $Date -StartTimeOfDay "13:30" -TimeZone 085 -DaysToRecur 3
```

### 2: Schedule a servicing window that occurs one day a week.

The first command gets the current date and adds seven days (one week), then stores the result in the \$Date variable.

The second command creates a servicing window named Test Servers Group 3 that occurs weekly on Saturday starting at 11:00 AM in the Eastern time zone and lasts for 3 hours (180 minutes). The start date of the servicing window is set to the date stored in \$Date, which is in seven days (one week).

```
PS C:\> $Date = (Get-Date).AddDays(7)
```

---

```
PS C:\> New-SCServicingWindow -Name "Test Servers Group 3" -Category "Test Group" -StartDate $Date -StartTimeOfDay "11:00" -TimeZone 035 -WeeklyScheduleDayOfWeek "Saturday" -WeeksToRecur 1 -MinutesDuration 180
```

### 3: Schedule a biweekly service window.

This command creates a servicing window named Staging Group C that occurs every other week (biweekly) on Saturday and Sunday starting at 10:30 PM in the Eastern time zone. Because no start date is specified, by default the servicing window becomes effective today.

```
PS C:\> New-SCServicingWindow -Name "Staging Group C" -StartTimeOfDay "22:30" -TimeZone 035 -WeeklyScheduleDayOfWeek "Saturday, Sunday" -WeeksToRecur 2
```

### 4: Schedule a bimonthly servicing window.

This command creates a servicing window named Production Servers A that occurs every other month (bimonthly) on the second Tuesday of the month, starting at 11:30 PM in the Eastern time zone. Because no start date is specified, by default the servicing window becomes effective today.

```
PS C:\> New-SCServicingWindow -Name "Production Servers A" -Category "Emergency" -StartTimeOfDay "23:30" -TimeZone 085 -MonthlyScheduleDayOfWeek "Tuesday" -WeekOfMonth "Second" -MonthsToRecur 2
```

## Related topics

[Get-SCServicingWindow](#)

[Remove-SCServicingWindow](#)

[Set-SCServicingWindow](#)



---

# New-SCSQLProfile

---

## New-SCSQLProfile

Creates a SQL Server profile.

### Syntax

Parameter Set: Default

```
New-SCSQLProfile [-Name] <String> [-Description <String> ] [-JobVariable <String> ] [-Owner <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-SQLProfile <SQLProfile> ] [-Tag <String> ] [-UserRole <UserRole> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The New-SQLProfile cmdlet creates a SQL Server profile.

### Parameters

#### **-Description<String>**

States a description for the specified object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Owner<String>**

Specifies the owner of a VMM object in the form of a valid domain user account.

Example format: -Owner "Contoso\ReneeLo"

Example format: -Owner "ReneeLo@Contoso"

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

---

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-SQLProfile<SQLProfile>**

Specifies a SQL Server profile object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

---

## **-Tag<String>**

Specifies a word or phrase to associate with an object so that you can search for all objects with the specified set of tags. You can search for a subset of tags, or you can search for the full set of tags.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-UserRole<UserRole>**

Specifies a user role object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **SQLProfile**

## Examples

### 1: Create a SQL Server profile.

The first command creates a SQL Server profile object named SQLProfile01 and stores the object in the \$SQLProfile variable.

The second command displays information about the SQL Server profile object stored in \$SQLProfile to the user.

```
PS C:\> $SQLProfile = New-SCSQLProfile -Name "SQLProfile01" -Description "SQL Profile 01" -Tag "Standard SQL Profile"
```

```
PS C:\> $SQLProfile
```

## Related topics

[Get-SCSQLProfile](#)

[Remove-SCSQLProfile](#)

[Set-SCSQLProfile](#)

---

# New-SCSSASConnection

---

## New-SCSSASConnection

Creates a connection to SQL Server Analysis Services.

### Syntax

Parameter Set: Default

```
New-SCSSASConnection -ComputerName <String> -Credential <VMMCredential> -InstanceName <String> -Port <Int32> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The New-SCSSASConnection cmdlet creates a connection to SQL Server Analysis Services (SSAS) and creates a database named VMMAnalysisServerDB on the SSAS server.

This cmdlet assumes that SSAS and Operations Manager Reporting Server are installed on the same computer. Therefore, New-SSASConnection also connects to the SQL Server Reporting Server Web Service to update the credentials for forecasting reports to connect to Analysis Services.

### Parameters

#### -ComputerName<String>

Specifies the name of a computer that VMM can uniquely identify on your network. Valid formats are: FQDN, IPv4 or IPv6 address, or NetBIOS name.

NOTE: See the examples for a specific cmdlet to determine how that cmdlet specifies the computer name.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-Credential<VMMSCredential>**

Specifies a credential object or, for some cmdlets, a Run As account object that contains the user name and password of an account that has permission to perform this action. Or, in the case of Restart-SCJob, has permission to complete a restarted task.

For more information about the PSCredential object, type: "Get-Help Get-Credential".

For more information about Run As accounts, type: "Get-Help New-SCRunAsAccount".

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-InstanceName<String>**

Specifies the SQL Server Analysis Services (SSAS) database instance name.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false

---

Accept Wildcard Characters?	false
-----------------------------	-------

### **-Port<Int32>**

Specifies the network port to use when adding an object or creating a connection. Valid values are: 1 to 4095.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none



Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **String**

## **Examples**

### **1: Create a connection to SQL Server Analysis Service.**

The first command gets the Run As account object named SSASRunAsAcct and stores the object in the \$SSASCreds variable. This credential needs to be an administrator in both the SQL Server Analysis Service Instance and SQL Server Reporting Instance.

The second command creates an SSAS connection to SQL Server SQLServer01, using the default instance name of MSSQLServer and port 2383. New-SCASSConnection uses the credentials provided in the Run As account stored in \$SSASCreds to create the connection.

```
PS C:\> $SSASCreds = Get-SCRunAsAccount -Name "SSASRunAsAcct"
```

```
PS C:\> $SSASSConnection = New-SCSSASConnection -ComputerName "SQLServer01" -InstanceName MSSQLServer -Port 2383 -Credential $SSASCreds
```

---

## Related topics

[Get-SCSSASConnection](#)

[Remove-SCSSASConnection](#)

---

# New-SCStaticIPAddressPool

---

## New-SCStaticIPAddressPool

Creates a static IP address pool.

### Syntax

Parameter Set: ByLogicalNetworkDefinition

```
New-SCStaticIPAddressPool -LogicalNetworkDefinition <LogicalNetworkDefinition> -Name <String> -Subnet <String> [-DefaultGateway <DefaultGateway[]> ] [-Description <String> ] [-DNSSearchSuffix <String[]> ] [-DNSServer <String[]> ] [-DNSSuffix <String> ] [-EnableNetBIOS <Boolean> ] [-IPAddressRangeEnd <String> ] [-IPAddressRangeStart <String> ] [-IPAddressReservedSet <String> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VIPAddressSet <String> ] [-Vlan <Int32> ] [-VMMServer <ServerConnection> ] [-WINSServer <String[]> ] [ <CommonParameters>]
```

Parameter Set: ByMulticastSwitch

```
New-SCStaticIPAddressPool -IsMulticast -LogicalNetworkDefinition <LogicalNetworkDefinition> -Name <String> -Subnet <String> [-Description <String> ] [-IPAddressRangeEnd <String> ] [-IPAddressRangeStart <String> ] [-IPAddressReservedSet <String> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ByVMSubnet

```
New-SCStaticIPAddressPool -Name <String> -Subnet <String> -VMSubnet <VMSubnet> [-DefaultGateway <DefaultGateway[]> ] [-Description <String> ] [-DNSSearchSuffix <String[]> ] [-DNSServer <String[]> ] [-DNSSuffix <String> ] [-EnableNetBIOS <Boolean> ] [-IPAddressRangeEnd <String> ] [-IPAddressRangeStart <String> ] [-IPAddressReservedSet <String> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VIPAddressSet <String> ] [-VMMServer <ServerConnection> ] [-WINSServer <String[]> ] [ <CommonParameters>]
```

### Detailed Description

The New-SCStaticIPAddressPool cmdlet creates a Virtual Machine Manager (VMM) static IP address pool. A static IP address pool can be associated with one or more host groups.

### Parameters

#### **-DefaultGateway<DefaultGateway[]>**

Specifies a default gateway object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Description<String>**

States a description for the specified object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-DNSSearchSuffix<String[]>**

Specifies one or more strings that are appended to a host name to resolve a DNS address.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-DNSServer<String[]>**

Specifies the IP address of one or more DNS servers. Valid formats: IPv4 or IPv6 address.

---

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-DNSSuffix<String>**

Specifies the default DNS suffix associated with a NIC.

Example format: -DNSSuffix "Domain01.Contoso.com"

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-EnableNetBIOS<Boolean>**

Indicates whether NetBIOS over TCP/IP is enabled for a NIC.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-IPAddressRangeEnd<String>**

Specifies the last IP address in a range of IP addresses. Use with IPAddressRangeStart.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-IPAddressRangeStart<String>**

Specifies the first IP address in a range of IP addresses. Use with IPAddressRangeEnd.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-IPAddressReservedSet<String>**

Specifies a set of IP addresses within an IP subnet that is reserved for other use.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-IsMulticast**

Indicates that the IP address is a multicast address or that the IP address pool contains a multicast IP address range.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-LogicalNetworkDefinition<LogicalNetworkDefinition>**

Specifies a logical network definition (also called a network site) that contains the subnet that the IP address pool serves as specified by the Subnet parameter.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false



---

## **-Subnet<String>**

Specifies an IP subnet (IPv4 or IPv6) in Classless Inter-Domain Routing (CIDR) notation.

Example format for an IPv4 subnet: 192.168.0.1/24

Example format for an IPv6 subnet: FD4A:29CD:184F:3A2C::/64

NOTE: An IP subnet cannot overlap with any other subnet in a host group or child host groups.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VIPAddressSet<String>**

Specifies a set of IP addresses within an IP subnet that is reserved for configuring virtual IPs (VIPs) in load balancers.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Vlan<Int32>**

Specifies a VLAN.

Aliases	none
Required?	false
Position?	named

Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-VMSubnet<VMSubnet>**

Specifies a VM subnet object.

To obtain a VM subnet object, use the Get-SCVMSubnet cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-WINS Server<String[]>**

Specifies the IP address of one or more WINS servers. Valid formats: IPv4 or IPv6 address.

Aliases	none
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## Examples

### 1: Create a Static IP Address Pool

The first command gets the host group with the path of All Hosts\HostGroup02\Production and stores it in the \$HostGroup variable.

The second command gets the logical network named LogicalNetwork01 and saves it in the \$LogNet variable.

The third command gets the logical network definition named Logical Network Definition 01 for the host group stored in the \$HostGroup variable.

The fourth command creates a new default gateway and stores it in the \$DefaultGateway variable.

The last command creates a new static IP address pool with the specified values.

```
PS C:\> $HostGroup = Get-SCVMHostGroup | where { $_.Path -eq "All
Hosts\HostGroup02\Production" }
PS C:\> $LogNet = Get-SCLogicalNetwork -Name "LogicalNetwork01"
PS C:\> $LogNetDef = Get-SCLogicalNetworkDefinition -VMHostGroup $HostGroup -LogicalNetwork
$LogNet -Name "Logical Network Definition 01"
PS C:\> $DefaultGateway = New-SCDefaultGateway -IPAddress "10.0.0.1" -Metric 10
PS C:\> New-SCStaticIPAddressPool -LogicalNetworkDefinition $LogNetDef -Name "Production IP
Address Pool" -Description "This IP address pool is used for LOB Apps in production" -Subnet
"10.0.0.0/24" -IPAddressRangeStart "10.0.0.10" -IPAddressRangeEnd "10.0.0.99" -
IPAddressReservedSet "10.0.0.25-10.0.0.35, 10.0.0.38" -VIPAddressSet "10.0.0.95-10.0.0.99" -
DNSSuffix "domain.contoso.com" -DNSSearchSuffix domain1.contoso.com, domain2.contoso.com -
DNSServer "10.0.0.1", "10.0.0.2" -WINSServer "10.0.0.1", "10.0.0.2" -DefaultGateway
$DefaultGateway -EnableNetBIOS $True
```

## Related topics

[Get-SCLogicalNetwork](#)

[Get-SCLogicalNetworkDefinition](#)

[Get-SCStaticIPAddressPool](#)

---

[Get-SCVMHostGroup](#)

[Remove-SCStaticIPAddressPool](#)

[Set-SCStaticIPAddressPool](#)

---

# New-SCStorageClassification

---

## New-SCStorageClassification

Creates a storage classification.

### Syntax

Parameter Set: Default

```
New-SCStorageClassification -Name <String> [-Description <String> ] [-JobVariable <String> ]  
[-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [  
<CommonParameters>]
```

### Detailed Description

The New-SCStorageClassification cmdlet creates a storage classification. A storage classification defines the capabilities of a storage pool. For example, a classification of Gold could be associated with storage pools that have the highest performance and availability.

For information about associating a storage classification with a storage pool, type: "Get-Help Set-SCStoragePool -examples".

### Parameters

#### -Description<String>

States a description for the specified object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **StorageClassification**

---

## Examples

### 1: Create a storage classification.

This command creates a storage classification named StorageClassification01 and stores it in the \$Class variable.

```
PS C:\> $Class = New-SCStorageClassification -Name "StorageClassification01" -Description  
"New storage classification"
```

### Related topics

[Get-SCStorageClassification](#)

[Remove-SCStorageClassification](#)

[Set-SCStorageClassification](#)



---

# New-SCStorageFileShare

---

## New-SCStorageFileShare

Creates a storage file share in VMM.

### Syntax

Parameter Set: NewShareOnNASFromHostedPool

```
New-SCStorageFileShare -HostedStoragePool <String> -Name <String> -SizeMB <UInt64> -StorageFileServer <StorageFileServer> [-Description <String> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: NewShareOnNASFromPool

```
New-SCStorageFileShare -Name <String> -SizeMB <UInt64> -StorageFileServer <StorageFileServer> -StoragePool <StoragePool> [-Description <String> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: NewShareOnWindowsServer

```
New-SCStorageFileShare -LocalPath <String> -Name <String> -StorageFileServer <StorageFileServer> [-ContinuouslyAvailable <Boolean> ] [-Description <String> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The New-SCStorageFileShare cmdlet creates a storage file share in Virtual Machine Manager (VMM).

### Parameters

#### **-ContinuouslyAvailable<Boolean>**

Indicates whether the file share is available on a continuous basis.

Aliases	none
Required?	false
Position?	named

---

Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Description<String>**

States a description for the specified object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-HostedStoragePool<String>**

Specifies a hosted storage pool.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
Required?	false
Position?	named

---

Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-LocalPath<String>**

Specifies a local path for a storage file share.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named

---

Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-SizeMB<UInt64>**

Specifies the size, in megabytes, of a storage file share.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-StorageFileServer<StorageFileServer>**

Specifies a storage file server object.

Aliases	none
Required?	true
Position?	named

Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-StoragePool<StoragePool>**

Specifies a storage pool object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

---

## Examples

### 1: Create a storage file share.

The first command gets the storage file server object named FileServer01 and stores the object in the \$FileServer variable.

The second command creates a storage file share named FileShare02 on FileServer01.

```
PS C:\> $FileServer = Get-SCStorageFileServer -Name "FileServer01.Contoso.com"
```

```
PS C:\> New-SCStorageFileShare -Name "FileShare02" -StorageFileServer $FileServer -LocalPath "C:\FileShare02"
```

## Related topics

[Get-SCStorageFileShare](#)

[Register-SCStorageFileShare](#)

[Remove-SCStorageFileShare](#)

[Repair-SCStorageFileShare](#)

[Set-SCStorageFileShare](#)

[Unregister-SCStorageFileShare](#)

---

# New-SCStorageLogicalUnit

---

## New-SCStorageLogicalUnit

Creates a logical unit from unallocated capacity in a storage pool.

### Syntax

Parameter Set: NewLunFromLun

```
New-SCStorageLogicalUnit [-StorageLogicalUnit] <StorageLogicalUnit> -Name <String> [-Description <String> ] [-JobVariable <String> ] [-LogicalUnitCopyMethod {Snapshot | Clone} ] [-PROTipID <Guid> ] [-RunAsynchronously] [-SetLogicalUnitCopySource] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: NewLunFromPool

```
New-SCStorageLogicalUnit [-StoragePool] <StoragePool> -DiskSizeMB <UInt64> -Name <String> [-Description <String> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-ProvisioningType {Unknown | Thin | Fixed} ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The New-SCStorageLogicalUnit cmdlet creates a logical unit from unallocated capacity in a storage pool.

### Parameters

#### -Description<String>

States a description for the specified object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

### **-DiskSizeMB<UInt64>**

Specifies, in megabytes (MB), the size of a disk.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-LogicalUnitCopyMethod<StorageLogicalUnitCopyMethod>**

Specifies the method used by the array to copy an existing logical unit. Valid values: Clone, Snapshot.

The acceptable values for this parameter are:

Snapshot	
Clone	

Aliases	none
Required?	false
Position?	named



---

Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-ProvisioningType<StorageProvisioningType>**

Specifies a storage provisioning type. Valid values are: Thin, Fixed.

Thin indicates that capacity is committed on demand for a storage logical unit.

Fixed indicates that capacity is fully committed for a fixed-size storage logical unit.

The acceptable values for this parameter are:

Unknown	
Thin	
Fixed	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-SetLogicalUnitCopySource**

Indicates that the specified storage logical unit is the source from which a clone is copied.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false

---

Accept Wildcard Characters?	false
-----------------------------	-------

### **-StorageLogicalUnit<StorageLogicalUnit>**

Specifies a storage logical unit object.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-StoragePool<StoragePool>**

Specifies a storage pool object.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)

---

Accept Wildcard Characters?
-----------------------------

false
-------

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **StorageLogicalUnit**

## Examples

### 1: Create a copy of an existing logical unit.

The first command gets the storage logical unit object named LUN01 and stores the object in the \$SourceLU variable.

The second command creates a new logical unit named NewLU by cloning LUN01.

```
PS C:\> $SourceLU = Get-SCStorageLogicalUnit -Name "LUN01"
```

```
PS C:\> New-SCStorageLogicalUnit -SetLogicalUnitCopySource -StorageLogicalUnit $SourceLU -  
Name "NewLU" -LogicalUnitCopyMethod Clone
```

## Related topics

[Get-SCStorageLogicalUnit](#)

[Register-SCStorageLogicalUnit](#)

[Remove-SCStorageLogicalUnit](#)

[Set-SCStorageLogicalUnit](#)

[Unregister-SCStorageLogicalUnit](#)

---

# New-SCSubnetVLAN

---

## New-SCSubnetVLAN

Creates a subnet VLAN object.

### Syntax

Parameter Set: Default

```
New-SCSubnetVLAN [-SecondaryVlanID <Int32> ] [-Subnet <String> ] [-SupportsDHCP <Boolean> ]  
[-VlanID <Int32> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The New-SCSubnetVLAN cmdlet creates a subnet VLAN object. You can use a subnet VLAN to create a logical network definition or VM subnet.

For information about how to create a logical network definition, type "Get-Help New-SCLogicalNetworkDefinition -detailed".

For information about how to create a VM subnet, type "Get-Help New-SCVMSubnet -detailed".

### Parameters

#### **-SecondaryVlanID<Int32>**

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

#### **-Subnet<String>**

Specifies an IP subnet (IPv4 or IPv6) in Classless Inter-Domain Routing (CIDR) notation.

Example format for an IPv4 subnet: 192.168.0.1/24

---

Example format for an IPv6 subnet: FD4A:29CD:184F:3A2C::/64

NOTE: An IP subnet cannot overlap with any other subnet in a host group or child host groups.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-SupportsDHCP<Boolean>**

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VlanID<Int32>**

Assigns a numerical identifier in the range 1-4094 to a virtual network adapter on a virtual machine or to a physical network adapter on a virtual machine host.

Configure a VlanID on a Hyper-V, VMware ESX, or Citrix XenServer host:

- On an externally bound physical network adapter when the Vlan mode is Access.

Configure a VlanID on a virtual network adapter of a virtual machine:

- Bound to a physical network adapter on the host, or
- Bound to an internal virtual network on the host.

Example format: -VlanEnabled

-VlanMode "Access" -VLANID 35

Aliases	none
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **SubnetVLAN**

## **Examples**

### **1: Create a subnet VLAN.**

This command creates a subnet VLAN object with a subnet value of 10.0.0.1/24 and a VLAN value of 25.

```
PS C:\> $SubnetVLAN = New-SCSubnetVLAN -Subnet 10.0.0.1/24 -VLAN 25
```

## **Related topics**

[New-SCLogicalNetworkDefinition](#)

---

# New-SCUplinkPortProfileSet

---

## New-SCUplinkPortProfileSet

Creates an uplink port profile set.

### Syntax

Parameter Set: Default

```
New-SCUplinkPortProfileSet [-Name] <String> [[-Description] <String> ] -LogicalSwitch <LogicalSwitch> [-ExtensionUplinkPortProfiles <ExtensionUplinkPortProfile[]> ] [-JobGroup <Guid> ] [-JobVariable <String> ] [-NativeUplinkPortProfile <NativeUplinkPortProfile> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters> ]
```

### Detailed Description

The New-SCUplinkPortProfileSet creates an uplink port profile set.

You must provide a logical switch when creating a uplink port profile set. To obtain a logical switch object, use the Get-SCLogicalSwitch cmdlet.

### Parameters

#### -Description<String>

States a description for the specified object.

Aliases	none
Required?	false
Position?	2
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false



---

## **-ExtensionUplinkPortProfiles<ExtensionUplinkPortProfile[]>**

Specifies an array of extension uplink port profiles.

To get an extension uplink port profile, use the Get-SCExtensionUplinkPortProfile cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **-JobGroup<Guid>**

Specifies an identifier for a series of commands that will run as a set just before the final command that includes the same job group identifier runs.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-LogicalSwitch<LogicalSwitch>**

Specifies a logical switch object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-NativeUplinkPortProfile<NativeUplinkPortProfile>**

Specifies a native uplink port profile object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

---

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## Examples

### 1: Create an uplink port profile set.

The first command gets the logical switch object named LogicalSwitch01 and stores the object in the \$LogSwitch variable.

The second command gets the native uplink port profile object named NativeUplinkPortProfile01 and stores the object in the \$NativeUplinkPortProf variable.

The last command creates an uplink port profile set named UplinkPortProfileSet01 with LogicalSwitch01 and NativeUplinkPortProfile01.

```
PS C:\> $LogSwitch = Get-SCLogicalSwitch -Name "LogicalSwitch01"
```

```
PS C:\> $NativeUplinkPortProf = Get-SCNativeUplinkPortProfile -Name  
"NativeUplinkPortProfile01"
```

```
PS C:\> New-SCUplinkPortProfileSet -Name "UplinkPortProfileSet01" -LogicalSwitch $LogSwitch  
-NativeUplinkPortProfile $NativeUplinkPortProf
```

## Related topics

[Get-SCLogicalSwitch](#)

[Get-SCUplinkPortProfileSet](#)

[Remove-SCUplinkPortProfileSet](#)

[Set-SCUplinkPortProfileSet](#)

---

# New-SCUserRole

---

## New-SCUserRole

Creates a user role for a group of VMM users.

### Syntax

Parameter Set: Default

```
New-SCUserRole [-Name] <String> -UserRoleProfile {Administrator | DelegatedAdmin | TenantAdmin | SelfServiceUser | ReadOnlyAdmin} [-Description <String> ] [-ID <Guid> ] [-JobGroup <Guid> ] [-JobVariable <String> ] [-ParentUserRole <UserRole> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The New-SCUserRole cmdlet creates a user role for a group of Virtual Machine Manager (VMM) users. You can create the following user roles: Delegated Administrator, Read-Only Administrator, and Self-Service User. Only one Administrator role exists; you cannot create another Administrator role or delete the existing one.

If you are a member of a Delegated Administrator user role, you can create a user role. However, the scope of the new user role must be a subset of the scope of its parent user role.

After you create a user role, you can use the Set-SCUserRole cmdlet to rename the user role, to add or remove members, and to add or modify the scope of objects that members of the role can manage. For a self-service user role, you can specify which actions members of a self-service user role can take on their virtual machines, and you can define a quota that limits the number of virtual machines self-service users can create. Although you cannot create or remove the Administrator role or limit its scope, you can use Set-SCUserRole to add or remove members to that role.

For information about setting the properties of a user role, type: "Get-Help Set-SCUserRole -detailed".

### Parameters

#### **-Description<String>**

States a description for the specified object.

Aliases	none
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-ID<Guid>**

Specifies the numerical identifier (as a globally unique identifier, or GUID) for a specific object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-JobGroup<Guid>**

Specifies an identifier for a series of commands that will run as a set just before the final command that includes the same job group identifier runs.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-ParentUserRole<UserRole>**

Specifies an existing VMM user role as the parent of a new user role.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-UserRoleProfile<Profile>**

Specifies the type of profile to use as the basis for the user role. Valid values are: DelegatedAdmin, ReadOnlyAdmin, SelfServiceUser.

The acceptable values for this parameter are:

Administrator	
DelegatedAdmin	
TenantAdmin	
SelfServiceUser	
ReadOnlyAdmin	

Aliases	none
Required?	true
Position?	named



Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **UserRole**

## **Examples**

### **1: Create a delegated administrator user role.**

This command creates a delegated administrator user role named "ContosoDelegatedAdmin", provides the description "Delegated Administrators for the Contoso.com domain", and uses the -UserRoleProfile parameter to designate the user role type as delegated administrator.

```
PS C:\> New-SCUserRole -Name "ContosoDelegatedAdmin" -Description "Delegated Administrators for the Contoso.com domain" -UserRoleProfile "DelegatedAdmin"
```

---

## 2: Create a Self Service User user role.

The this command creates a new user role named "ContosoSelfServiceUsers", uses the -UserRoleProfile parameter to designate the new user role type as Self Service User, and stores the new user role object in the \$SelfServiceRole variable.

```
PS C:\> $SelfServiceRole = New-SCUserRole -Name "ContosoSelfServiceUsers" -UserRoleProfile "SelfServiceUser"
```

### Related topics

[Get-SCUserRole](#)

[Remove-SCUserRole](#)

[Set-SCUserRole](#)

---

# New-SCV2V

---

## New-SCV2V

Converts a virtual machine created on a VMware ESX Server host to a virtual machine deployed on a Hyper-V host managed by VMM.

### Syntax

Parameter Set: FromVM

```
New-SCV2V -VM <VM> -VMHost <Host> [-CPUCount <Byte> ] [-CPURelativeWeight <Int32> ] [-CPUType <ProcessorType> ] [-DelayStartSeconds <Int32> ] [-Description <String> ] [-EnableMACAddressSpoofing <Boolean> ] [-EnableVMNetworkOptimization <Boolean> ] [-JobGroup <Guid> ] [-JobVariable <String> ] [-LogicalNetwork <LogicalNetwork> ] [-MACAddress <String> ] [-MACAddressType <String> ] [-MemoryMB <Int32> ] [-Name <String> ] [-NetworkLocation <String> ] [-NetworkTag <String> ] [-NoConnection] [-OverridePatchPath <String> ] [-Owner <String> ] [-Path <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-SkipInstallVirtualizationGuestServices] [-SourceNetworkConnectionID <String> ] [-StartAction {NeverAutoTurnOnVM | AlwaysAutoTurnOnVM | TurnOnVMIIfRunningWhenVSSStopped} ] [-StartVM] [-StopAction {SaveVM | TurnOffVM | ShutdownGuestOS} ] [-Trigger] [-UserRole <UserRole> ] [-VirtualNetwork <VirtualNetwork> ] [-VirtualNetworkAdapter <VirtualNetworkAdapter> ] [-VLANEnabled <Boolean> ] [-VLANID <UInt16> ] [-VMMServer <ServerConnection> ] [-VMNetwork <VMNetwork> ] [ <CommonParameters>]
```

Parameter Set: NoParse

```
New-SCV2V -VMHost <Host> -VMXComputerConfiguration <VmxMachineConfiguration> [-CPUCount <Byte> ] [-CPURelativeWeight <Int32> ] [-CPUType <ProcessorType> ] [-DelayStartSeconds <Int32> ] [-Description <String> ] [-EnableMACAddressSpoofing <Boolean> ] [-EnableVMNetworkOptimization <Boolean> ] [-JobGroup <Guid> ] [-JobVariable <String> ] [-LibraryServer <LibraryServer> ] [-LogicalNetwork <LogicalNetwork> ] [-MACAddress <String> ] [-MACAddressType <String> ] [-MemoryMB <Int32> ] [-Name <String> ] [-NetworkLocation <String> ] [-NetworkTag <String> ] [-NoConnection] [-OverridePatchPath <String> ] [-Owner <String> ] [-Path <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-SkipInstallVirtualizationGuestServices] [-SourceNetworkConnectionID <String> ] [-StartAction {NeverAutoTurnOnVM | AlwaysAutoTurnOnVM | TurnOnVMIIfRunningWhenVSSStopped} ] [-StartVM] [-StopAction {SaveVM | TurnOffVM | ShutdownGuestOS} ] [-Trigger] [-UserRole <UserRole> ] [-VirtualNetwork <VirtualNetwork> ] [-VirtualNetworkAdapter <VirtualNetworkAdapter> ] [-VLANEnabled <Boolean> ] [-VLANID <UInt16> ] [-VMMServer <ServerConnection> ] [-VMNetwork <VMNetwork> ] [ <CommonParameters>]
```

Parameter Set: Parse

```
New-SCV2V -VMHost <Host> -VMXPath <String> [-CPUCount <Byte> ] [-CPURelativeWeight <Int32> ] [-CPUType <ProcessorType> ] [-DelayStartSeconds <Int32> ] [-Description <String> ] [-EnableMACAddressSpoofing <Boolean> ] [-EnableVMNetworkOptimization <Boolean> ] [-JobGroup <Guid> ] [-JobVariable <String> ] [-LibraryServer <LibraryServer> ] [-LogicalNetwork <LogicalNetwork> ] [-MACAddress <String> ] [-MACAddressType <String> ] [-MemoryMB <Int32> ] [-Name <String> ] [-NetworkLocation <String> ] [-NetworkTag <String> ] [-NoConnection] [-OverridePatchPath <String> ] [-Owner <String> ] [-Path <String> ] [-PROTipID <Guid> ] [-
```

---

```
RunAsynchronously] [-SkipInstallVirtualizationGuestServices] [-SourceNetworkConnectionID
<String> ] [-StartAction {NeverAutoTurnOnVM | AlwaysAutoTurnOnVM |
TurnOnVMIfRunningWhenVSSStopped} ] [-StartVM] [-StopAction {SaveVM | TurnOffVM |
ShutdownGuestOS} ] [-Trigger] [-UserRole <UserRole> ] [-VirtualNetwork <VirtualNetwork> ] [-
VirtualNetworkAdapter <VirtualNetworkAdapter> ] [-VlanEnabled <Boolean> ] [-VlanID <UInt16>
] [-VMMServer <ServerConnection> ] [-VMNetwork <VMNetwork> ] [ <CommonParameters>]
```

## Detailed Description

The New-SCV2V cmdlet converts a virtual machine created on a VMware ESX Server host to a virtual machine deployed on a Hyper-V host managed by Virtual Machine Manager (VMM). You cannot specify a VMware ESX host as the destination host for the new virtual machine.

### VMM V2V CONVERSION REQUIREMENTS

-----

A V2V conversion requires that the host on which the new virtual machine will be deployed is a Hyper-V host.

The source for a V2V conversion of a VMware virtual machine performed by New-SCV2V is a set of files that you must store in the VMM library before you perform the conversion:

- A .vmx file, which is a VMware virtual machine configuration file.

A .vmx file is approximately similar in function to the virtual machine configuration file (.vmc file) used for a Windows-based virtual machine.

A .vmx file is a text file that describes the properties and structure of a virtual machine, including name, memory, disk assignments, network parameters, and so on.

- One or more .vmdk files. A .vmdk file is a VMware virtual hard disk file, which is similar to the virtual hard disk file (.vhd file) used for a Windows-based virtual machine.

The .vmdk files are not passed directly as input to New-SCV2V but are listed in the .vmx file. A .vmdk file contains the virtual machine's guest operating system, applications, and data.

Supported VMware virtual hard disk formats include:

- monolithicSparse
- monolithicFlat
- vmfs
- twoGbMaxExtentSparse
- twoGbMaxExtentFlat

### VMM V2V CONVERSION PROCESS

-----

---

During the conversion process, New-SCV2V converts the .vmdk files to .vhd files and makes the operating system on the new virtual machine compatible with Hyper-V. The virtual machine created by New-SCV2V matches VMware virtual machine properties, including name, description, memory, disk-to-bus assignment, and so on, unless these settings are explicitly overridden by specifying different values for these settings. By default, the conversion process does not preserve network adapter settings; however, you can explicitly set adapter settings on the target virtual machine.

#### VMM V2V CONVERSION OF THE GUEST OPERATING SYSTEM

-----

New-SCV2V supports the conversion of VMware virtual machines that are running any of the following guest operating systems:

- \* Microsoft Windows 2000 Server with Service Pack 4 (SP4) or later
- \* Windows Server 2003 SP1 or later
- \* Windows Server 2003 R2 or later
- \* Windows Server 2008 or later
- \* Windows XP SP1 or later
- \* Windows Vista

Some conversions of a VMware-based virtual machine whose guest operating system is Windows might require that additional system files and drivers be added to the internal cache. You can use the Add-SCPatch cmdlet to add the required files to the cache. To determine what patches you need to add, run New-SCV2V and let the cmdlet convert the .vmdk file to a .vhd file. If you need patches, this process will put the V2V conversion into a failed state and will produce a list of required patches. Next, use the Add-SCPatch cmdlet to add the patches to the internal cache, and then restart the failed V2V job. The V2V process will continue and will not need to redo the disk conversion.

If you use New-SCV2V to convert a VMware-based virtual machine running any other operating system to a Hyper-V or Virtual Server-based virtual machine, the virtual machine might not start up or might not function correctly. To ensure a successful conversion, you must first modify the guest operating system to one of the listed supported operating systems.

#### FOR MORE INFORMATION

-----

- About how VMM can convert VMDK files directly, type: "Get-Help Copy-VirtualHardDisk".
- About how to add required files to the internal cache, type: "Get-Help Add-SCPatch".

## Parameters

### **-CPUCount<Byte>**

Specifies the number of CPUs on a virtual machine, on a hardware profile, or on a template. See the examples for a specific cmdlet to determine how that cmdlet uses this parameter.

TYPE OF HOST    NUMBER OF PROCESSORS

-----    -----

Hyper-V        Up to 4 CPUs per VM; varies by guest OS

VMware ESX    Up to 4 CPUs per VM for any supported guest OS

Exception: 1 CPU on a VM running Windows NT 4.0  
 Citrix XenServer Up to 8 CPUs per VM; varies by guest OS

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-CPURelativeWeight<Int32>**

Specifies the amount of CPU resources on a host that this virtual machine can use relative to other virtual machines on the same host. A virtual machine with a higher setting is allocated more CPU resources than a virtual machine with a lower setting.

TYPE OF HOST    RANGE OF RELATIVE VALUES

-----    -----

Hyper-V        1 to 10000

VMware ESX    2000 = High

1500 = Above Normal

1000 = Normal (default)

750 = Below Normal

500 = Low

1 to 1000000 = Custom

The VMware term for these values is "shares."

Citrix XenServer 1 to 65536, normal is 256

NOTE: See the examples for a specific cmdlet to determine how that cmdlet uses this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-CPUType<ProcessorType>**

Specifies the type of CPU for a virtual machine. To retrieve a list of all CPU types that are available for use in virtual machines in a VMM environment, type: "Get-SCCPUType"

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-DelayStartSeconds<Int32>**

Specifies the number of seconds to wait after the virtualization service starts before automatically starting a virtual machine. This delay is used to stagger the startup time of multiple virtual machines to help reduce the demand on the physical computer's resources. A typical setting might be 30 to 60 seconds.

TYPE OF HOST      MAXIMUM CONFIGURABLE DELAY

-----

Hyper-V      1000000000 seconds (277777 hours)

VMware ESX      65535 seconds (18 hours)

Citrix XenServer      Does not apply to XenServer virtual machines

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Description<String>**

States a description for the specified object.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-EnableMACAddressSpoofing<Boolean>**

Enables, when set to \$True, MAC Address spoofing.

Aliases	MACAddressesSpoofingEnabled
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-EnableVMNetworkOptimization<Boolean>**

Enables, when set to \$True, virtual machine network optimization. This feature improves network performance for virtual machines with network adapters that support virtual machine queue (VMQ) or TCP Chimney Offload. VMQ enables creating a unique network queue for each virtual network adapter. TCP Chimney Offload enables network traffic processing to be offloaded from the networking stack.

Aliases	VMNetworkOptimizationEnabled
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false



---

## **-JobGroup<Guid>**

Specifies an identifier for a series of commands that will run as a set just before the final command that includes the same job group identifier runs.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-LibraryServer<LibraryServer>**

Specifies a VMM library server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-LogicalNetwork<LogicalNetwork>**

Specifies a logical network. A logical network is a named grouping of IP subnets and VLANs that is used to organize and simplify network assignments.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-MACAddress<String>**

Specifies the Media Access Control (MAC) address, or a set of MAC addresses, for a physical or virtual network adapter on a computer. Valid values are: one or more MAC addresses.

Example format for a single MAC address:

`-MACAddress "00-15-5D-B4-DC-00"`

Example format for a set of MAC addresses:

`-MACAddress "00-15-5D-B4-DC-00", "00-1A-A0-E3-75-29"`

Example format for a set of MAC addresses:

`$Macs = "00-15-5D-B4-DC-00", "00-1A-A0-E3-75-29"`

`Set-SCPXEServer -MACAddress $Macs`

NOTE: When used with `New-SCPXEServer` or `Set-SCPXEServer`, the `MACAddress` parameter updates the PXE interfaces from which the SCDM PXE Server listens for and responds to PXE requests

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-MACAddressType<String>**

Specifies the type of MAC address to use for a virtual network adapter. Valid values are: Static, Dynamic.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-MemoryMB<Int32>**

Specifies, in megabytes (MB), the amount of random access memory (RAM) on the host that is allocated to a virtual machine. The default value is 512 MB. For a virtual machine on which dynamic memory is enabled (on a host running Windows Server 2008 R2 SP1 or later), use MemoryMB to specify the startup memory value.

TYPE OF HOST                      MAXIMUM HOST MEMORY ASSIGNABLE TO VM

-----                      -----

Hyper-V                      Up to 65536 MB RAM per virtual machine

VMware ESX Server 3.0.x Up to 16384 MB RAM per virtual machine

VMware ESX Server 3.5.x Up to 65532 MB RAM per virtual machine

Citrix XenServer    Up to 32265 MB RAM per VM

Example format: -MemoryMB 1024

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Name<String>**

Specifies the name of a VMM object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-NetworkLocation<String>**

Specifies the network location for a physical network adapter or for a virtual network adapter, or changes the default network location of a host's physical network adapter.

Example formats:

-NetworkLocation \$NetLoc (\$NetLoc might contain "Corp.Contoso.com")

-OverrideNetworkLocation \$TRUE -NetworkLocation "HostNICNewLocation.Contoso.com"

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-NetworkTag<String>**

Specifies a word or phrase to associate with a virtual network adapter that is configured to connect to a specific internal or external network on the host. The NetworkTag identifies all virtual machines with the same NetworkTag as members of the same network. VMM uses a NetworkTag (if one exists) when it evaluates hosts as possible candidates on which to deploy a virtual machine. If the host does not include virtual machines on the network with the same NetworkTag as the virtual machine to be placed, the host receives zero stars in the placement process.

Aliases	none
Required?	false
Position?	named

Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-NoConnection**

Disconnects a virtual network adapter from a virtual network.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-OverridePatchPath<String>**

For internal use only (not for use in your code).

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Owner<String>**

Specifies the owner of a VMM object in the form of a valid domain user account.

Example format: -Owner "Contoso\ReneeLo"

Example format: -Owner "ReneeLo@Contoso"

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Path<String>**

Specifies the destination path for the operation.

Example formats:

Local path -Path "F:\"

UNC path -Path "\\Library\Templates"

Volume GUID path -Path "\\?\Volume{4703c1ea-8ae7-11db-b473-00123f7603e3}\"

VMware ESX path -Path "[storage1]\MyVMwareFolderForVMs\MyVM.vmx"

Citrix XenServer path - Path "Local storage[99b6212f-b63d-c676-25f9-d6c460992de7]"

Wildcards are supported for "Get" cmdlets and when you specify the UNC path:

Example format:

UNC path -Path "\\VMHostServer\MyVMs\\*VM\*"

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-SkipInstallVirtualizationGuestServices**

Skips the installation of virtualization guest services on a virtual machine. By default, this parameter is set to \$False and VMM installs the appropriate virtualization guest service automatically. For a virtual machine on a Hyper-V host, the virtualization guest service is called Integration Components (VMGuest.iso). For a virtual machine on a XenServer host, the virtualization guest service is called Citrix Tools for Virtual Machines (xs-tools.iso). Virtual machines on a VMware ESX host do not use a virtualization guest service.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-SourceNetworkConnectionID<String>**

Specifies the MAC address or network name of the physical network adapter to be converted into a virtual network adapter in the virtual machine.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-StartAction<VMStartAction>**

Specifies the behavior of a virtual machine when the virtualization service (Hyper-V, VMware, or XenServer) starts. Valid values are: AlwaysAutoTurnOnVM, NeverAutoTurnOnVM, TurnOnVMIfRunningWhenVSStopped.

The acceptable values for this parameter are:

NeverAutoTurnOnVM	
AlwaysAutoTurnOnVM	
TurnOnVMIfRunningWhenVSStopped	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-StartVM**

Specifies that the virtual machine starts when it arrives at the destination host.

Aliases	none
Required?	false
Position?	named



Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-StopAction<VMStopAction>**

Specifies the behavior of the virtual machine when the virtualization service (Hyper-V, VMware, or XenServer) stops. Valid values are: SaveVM, TurnOffVM, ShutdownGuestOS.

The acceptable values for this parameter are:

SaveVM	
TurnOffVM	
ShutdownGuestOS	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Trigger**

Starts running the commands in a job group for a physical-to-virtual (P2V) conversion, a virtual-to-virtual (V2V) conversion, or the conversion of a physical hard disk to a virtual hard disk.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-UserRole<UserRole>**

Specifies a user role object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VirtualNetwork<VirtualNetwork>**

Specifies a virtual network object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VirtualNetworkAdapter<VirtualNetworkAdapter>**

Specifies a virtual network adapter object for a virtual machine.

TYPE OF HOST      NUMBER OF VIRTUAL NETWORK ADAPTERS

-----      -----

Hyper-V            Up to 4 emulated adapters per virtual machine.

Up to 8 synthetic adapters per virtual machine.

(Exception: no driver available for an emulated network adapter on a Windows Server 2003 x64 guest.)

VMware ESX        Up to 4 emulated adapters per virtual machine.

Citrix XenServer   Up to 7 emulated adapters per virtual machine.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VlanEnabled<Boolean>**

Enables a virtual LAN (VLAN) for use by virtual machines on a Hyper-V or Citrix XenServer host.

Example format for a single VLAN: -VLANEnabled -VLANMode "Access" -VLANID 35

Example format for multiple VLANs: -VLANEnabled -VLANMode "Trunk" -VLANTrunkID 1,2,100,200,1124

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VlanID<UInt16>**

Assigns a numerical identifier in the range 1-4094 to a virtual network adapter on a virtual machine or to a physical network adapter on a virtual machine host.

Configure a VlanID on a Hyper-V, VMware ESX, or Citrix XenServer host:

- On an externally bound physical network adapter when the Vlan mode is Access.

Configure a VlanID on a virtual network adapter of a virtual machine:

- Bound to a physical network adapter on the host, or
- Bound to an internal virtual network on the host.

Example format: -VlanEnabled

-VlanMode "Access" -VLANID 35

Aliases	none
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VM<VM>**

Specifies a virtual machine object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-VMHost<Host>**

Specifies a virtual machine host object. VMM supports Hyper-V hosts, VMware ESX hosts, and Citrix XenServer hosts.

For more information about each type of host, type: "Get-Help Add-SCVMHost -detailed". See the examples for a specific cmdlet to determine how that cmdlet uses this parameter.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-VMNetwork<VMNetwork>**

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VMXComputerConfiguration<VmxMachineConfiguration>**

Specifies a VMX computer configuration object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-VMXPath<String>**

Specifies the full UNC path to the .vmx file of a VMware virtual machine.

Example format: \\ServerName\VolumeName\DirectoryName\VMwareVM.vmx

---

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **VirtualMachine**

## Notes

- Requires a VMM VMX computer configuration object, which can be retrieved by using the Get-SCVMXComputerConfiguration cmdlet.

## Examples

### 1: Convert a VMware-based virtual machine deployed on an ESX host to a virtual machine deployed on a Hyper-V host.

The first command gets the host object named ESXHost01 and stores the object in the \$ESXHost variable.

The second command gets the host object named HyperVHost01 in the Contoso.com domain and stores the object in the \$VMHost variable.

The third command gets the virtual machine object SourceVM on ESXHost01 and stores the object in the \$VM variable.

In the last command, New-SCV2V performs the following operations:

- Creates a Windows-based virtual machine named DestinationVM from the source VMware virtual machine named SourceVM. The command deploys the new virtual machine, now named DestinationVM, onto HyperVHost01, storing the virtual machine files in the folder C:\VMs on HyperVHost01.
- Assigns 512 MB of memory on HyperVHost01 for use by the new virtual machine.

---

- Uses the `-RunAsynchronously` parameter to return control to the command shell immediately, before the command completes.

All of the virtual disks on the source virtual machine will be converted and attached to the new virtual machine.

```
PS C:\> $ESXHost = Get-SCVMHost -ComputerName "ESXHost01"
PS C:\> $VMHost = Get-SCVMHost -ComputerName "HyperVHost01.Contoso.com"
PS C:\> $VM = Get-SCVirtualMachine -VMHost $ESXHost -Name "SourceVM"
PS C:\> New-SCV2V -VM $VM -VMHost $VMHost -Name "DestinationVM" -Path "C:\VMs" -MemoryMB 512
-RunAsynchronously
```

## 2: Convert a VMware-based virtual machine stored in the VMM library to a virtual machine deployed on a Hyper-V host.

The first command gets the library server object named `LibServer02` and stores the object in the `$LibServ` variable.

The second command gets the host object named `VirtualServerHost02` and stores the object in the `$VMHost` variable.

In the last command, `New-SCV2V` performs the following operations:

- Creates a Windows-based virtual machine named `VM02` from the source VMware file (`VMSource.vmx`) stored at the specified path on `FileServer02`, and then deploys the new virtual machine (`VM02`) onto `VirtualServerHost02`. The command stores the virtual machine files in the folder `C:\VMs` on `VirtualServerHost02`.
- Assigns 512 MB of memory on `VirtualServerHost02` for use by the new virtual machine.
- Uses the `-RunAsynchronously` parameter to return control to the command shell immediately, before the command completes.

```
PS C:\> $LibServ = Get-SCLibraryServer -ComputerName "LibServer02.Contoso.com"
PS C:\> $VMHost = Get-SCVMHost -ComputerName "VirtualServerHost02.Contoso.com"
PS C:\> New-SCV2V -LibraryServer $LibServ -VMXPath
"\\LibServer02\MSSCVMMLibrary\VMware\VMSource.vmx" -VMHost $VMHost -Name "VM02" -Path
"C:\VMs" -MemoryMB 512 -RunAsynchronously
```

## Related topics

[Add-SCPatch](#)

[Copy-SCStorageVolume](#)

[Copy-SCVirtualHardDisk](#)

[New-SCVMXComputerConfiguration](#)

[Remove-SCComputerConfiguration](#)

---

# New-SCVirtualDiskDrive

---

## New-SCVirtualDiskDrive

Creates a virtual disk drive object on a virtual machine deployed on a host managed by VMM, or on a template in the VMM library.

### Syntax

Parameter Set: AnyHostDiskToJobGroupParamSetIDE

```
New-SCVirtualDiskDrive -AnyStorageDisk -Bus <Byte> -IDE -JobGroup <Guid> -LUN <Byte> [-BootVolume] [-JobVariable <String> ] [-PROTipID <Guid> ] [-ReturnImmediately] [-RunAsynchronously] [-StorageClassification <StorageClassification> ] [-SystemVolume] [-VirtualHardDiskFormatType {VHD | VMDK | VHDX} ] [-VMMServer <ServerConnection> ] [-VolumeType {None | Boot | System | BootAndSystem} ] [ <CommonParameters>]
```

Parameter Set: AnyHostDiskToJobGroupParamSetSCSI

```
New-SCVirtualDiskDrive -AnyStorageDisk -Bus <Byte> -JobGroup <Guid> -LUN <Byte> -SCSI [-BootVolume] [-JobVariable <String> ] [-PROTipID <Guid> ] [-ReturnImmediately] [-RunAsynchronously] [-StorageClassification <StorageClassification> ] [-SystemVolume] [-VirtualHardDiskFormatType {VHD | VMDK | VHDX} ] [-VMMServer <ServerConnection> ] [-VolumeType {None | Boot | System | BootAndSystem} ] [ <CommonParameters>]
```

Parameter Set: AnyHostDiskToTemplateParamSetIDE

```
New-SCVirtualDiskDrive -AnyStorageDisk -Bus <Byte> -IDE -LUN <Byte> -VMTemplate <Template> [-BootVolume] [-JobVariable <String> ] [-PROTipID <Guid> ] [-ReturnImmediately] [-RunAsynchronously] [-StorageClassification <StorageClassification> ] [-SystemVolume] [-VirtualHardDiskFormatType {VHD | VMDK | VHDX} ] [-VMMServer <ServerConnection> ] [-VolumeType {None | Boot | System | BootAndSystem} ] [ <CommonParameters>]
```

Parameter Set: AnyHostDiskToTemplateParamSetSCSI

```
New-SCVirtualDiskDrive -AnyStorageDisk -Bus <Byte> -LUN <Byte> -SCSI -VMTemplate <Template> [-BootVolume] [-JobVariable <String> ] [-PROTipID <Guid> ] [-ReturnImmediately] [-RunAsynchronously] [-StorageClassification <StorageClassification> ] [-SystemVolume] [-VirtualHardDiskFormatType {VHD | VMDK | VHDX} ] [-VMMServer <ServerConnection> ] [-VolumeType {None | Boot | System | BootAndSystem} ] [ <CommonParameters>]
```

Parameter Set: existingHostVHDTtoVMPParamSetIDE

```
New-SCVirtualDiskDrive -Bus <Byte> -FileName <String> -IDE -LUN <Byte> -Path <String> -UseLocalVirtualHardDisk -VM <VM> [-BootVolume] [-JobVariable <String> ] [-PROTipID <Guid> ] [-ReturnImmediately] [-RunAsynchronously] [-StorageClassification <StorageClassification> ] [-SystemVolume] [-VirtualHardDiskFormatType {VHD | VMDK | VHDX} ] [-VMMServer <ServerConnection> ] [-VolumeType {None | Boot | System | BootAndSystem} ] [ <CommonParameters>]
```

Parameter Set: existingHostVHDTtoVMPParamSetIDEJobGroup

```
New-SCVirtualDiskDrive -Bus <Byte> -FileName <String> -IDE -JobGroup <Guid> -LUN <Byte> -Path <String> -UseLocalVirtualHardDisk [-BootVolume] [-JobVariable <String> ] [-PROTipID
```



---

<Guid> ] [-ReturnImmediately] [-RunAsynchronously] [-StorageClassification  
<StorageClassification> ] [-SystemVolume] [-VirtualHardDiskFormatType {VHD | VMDK | VHDX} ]  
[-VMMServer <ServerConnection> ] [-VolumeType {None | Boot | System | BootAndSystem} ] [  
<CommonParameters>]

Parameter Set: existingHostVHDToVMPParamSetSCSI

New-SCVirtualDiskDrive -Bus <Byte> -FileName <String> -LUN <Byte> -Path <String> -SCSI -  
UseLocalVirtualHardDisk -VM <VM> [-BootVolume] [-JobVariable <String> ] [-PROTipID <Guid> ]  
[-ReturnImmediately] [-RunAsynchronously] [-StorageClassification <StorageClassification> ]  
[-SystemVolume] [-VirtualHardDiskFormatType {VHD | VMDK | VHDX} ] [-VMMServer  
<ServerConnection> ] [-VolumeType {None | Boot | System | BootAndSystem} ] [  
<CommonParameters>]

Parameter Set: existingHostVHDToVMPParamSetSCSIJobGroup

New-SCVirtualDiskDrive -Bus <Byte> -FileName <String> -JobGroup <Guid> -LUN <Byte> -Path  
<String> -SCSI -UseLocalVirtualHardDisk [-BootVolume] [-JobVariable <String> ] [-PROTipID  
<Guid> ] [-ReturnImmediately] [-RunAsynchronously] [-StorageClassification  
<StorageClassification> ] [-SystemVolume] [-VirtualHardDiskFormatType {VHD | VMDK | VHDX} ]  
[-VMMServer <ServerConnection> ] [-VolumeType {None | Boot | System | BootAndSystem} ] [  
<CommonParameters>]

Parameter Set: existingVHDToJobGroupParamSetIDE

New-SCVirtualDiskDrive -Bus <Byte> -IDE -JobGroup <Guid> -LUN <Byte> -VirtualHardDisk  
<StandaloneVirtualHardDisk> [-BootVolume] [-FileName <String> ] [-JobVariable <String> ] [-  
Path <String> ] [-PROTipID <Guid> ] [-ReturnImmediately] [-RunAsynchronously] [-  
StorageClassification <StorageClassification> ] [-SystemVolume] [-VirtualHardDiskFormatType  
{VHD | VMDK | VHDX} ] [-VMMServer <ServerConnection> ] [-VolumeType {None | Boot | System |  
BootAndSystem} ] [ <CommonParameters>]

Parameter Set: existingVHDToJobGroupParamSetSCSI

New-SCVirtualDiskDrive -Bus <Byte> -JobGroup <Guid> -LUN <Byte> -SCSI -VirtualHardDisk  
<StandaloneVirtualHardDisk> [-BootVolume] [-FileName <String> ] [-JobVariable <String> ] [-  
Path <String> ] [-PROTipID <Guid> ] [-ReturnImmediately] [-RunAsynchronously] [-  
StorageClassification <StorageClassification> ] [-SystemVolume] [-VirtualHardDiskFormatType  
{VHD | VMDK | VHDX} ] [-VMMServer <ServerConnection> ] [-VolumeType {None | Boot | System |  
BootAndSystem} ] [ <CommonParameters>]

Parameter Set: existingVHDToTemplateParamSetIDE

New-SCVirtualDiskDrive -Bus <Byte> -IDE -LUN <Byte> -VirtualHardDisk  
<StandaloneVirtualHardDisk> -VMTemplate <Template> [-BootVolume] [-JobVariable <String> ] [-  
PROTipID <Guid> ] [-ReturnImmediately] [-RunAsynchronously] [-StorageClassification  
<StorageClassification> ] [-SystemVolume] [-VirtualHardDiskFormatType {VHD | VMDK | VHDX} ]  
[-VMMServer <ServerConnection> ] [-VolumeType {None | Boot | System | BootAndSystem} ] [  
<CommonParameters>]

Parameter Set: existingVHDToTemplateParamSetSCSI

New-SCVirtualDiskDrive -Bus <Byte> -LUN <Byte> -SCSI -VirtualHardDisk  
<StandaloneVirtualHardDisk> -VMTemplate <Template> [-BootVolume] [-JobVariable <String> ] [-  
PROTipID <Guid> ] [-ReturnImmediately] [-RunAsynchronously] [-StorageClassification  
<StorageClassification> ] [-SystemVolume] [-VirtualHardDiskFormatType {VHD | VMDK | VHDX} ]  
[-VMMServer <ServerConnection> ] [-VolumeType {None | Boot | System | BootAndSystem} ] [  
<CommonParameters>]

---

Parameter Set: existingVHDToVMPParamSetIDE

New-SCVirtualDiskDrive -Bus <Byte> -IDE -LUN <Byte> -VirtualHardDisk <StandaloneVirtualHardDisk> -VM <VM> [-BootVolume] [-FileName <String> ] [-JobVariable <String> ] [-Path <String> ] [-PROTipID <Guid> ] [-ReturnImmediately] [-RunAsynchronously] [-StorageClassification <StorageClassification> ] [-SystemVolume] [-VirtualHardDiskFormatType {VHD | VMDK | VHDX} ] [-VMMServer <ServerConnection> ] [-VolumeType {None | Boot | System | BootAndSystem} ] [ <CommonParameters>]

Parameter Set: existingVHDToVMPParamSetSCSI

New-SCVirtualDiskDrive -Bus <Byte> -LUN <Byte> -SCSI -VirtualHardDisk <StandaloneVirtualHardDisk> -VM <VM> [-BootVolume] [-FileName <String> ] [-JobVariable <String> ] [-Path <String> ] [-PROTipID <Guid> ] [-ReturnImmediately] [-RunAsynchronously] [-StorageClassification <StorageClassification> ] [-SystemVolume] [-VirtualHardDiskFormatType {VHD | VMDK | VHDX} ] [-VMMServer <ServerConnection> ] [-VolumeType {None | Boot | System | BootAndSystem} ] [ <CommonParameters>]

Parameter Set: HostDiskToJobGroupParamSetIDE

New-SCVirtualDiskDrive -Bus <Byte> -IDE -JobGroup <Guid> -LUN <Byte> -StorageDisk <StorageDisk> [-BootVolume] [-JobVariable <String> ] [-PROTipID <Guid> ] [-ReturnImmediately] [-RunAsynchronously] [-StorageClassification <StorageClassification> ] [-SystemVolume] [-VirtualHardDiskFormatType {VHD | VMDK | VHDX} ] [-VMMServer <ServerConnection> ] [-VolumeType {None | Boot | System | BootAndSystem} ] [ <CommonParameters>]

Parameter Set: HostDiskToJobGroupParamSetSCSI

New-SCVirtualDiskDrive -Bus <Byte> -JobGroup <Guid> -LUN <Byte> -SCSI -StorageDisk <StorageDisk> [-BootVolume] [-FileName <String> ] [-JobVariable <String> ] [-Path <String> ] [-PROTipID <Guid> ] [-ReturnImmediately] [-RunAsynchronously] [-StorageClassification <StorageClassification> ] [-SystemVolume] [-VirtualHardDiskFormatType {VHD | VMDK | VHDX} ] [-VMMServer <ServerConnection> ] [-VolumeType {None | Boot | System | BootAndSystem} ] [ <CommonParameters>]

Parameter Set: HostDiskToVMPParamSetIDE

New-SCVirtualDiskDrive -Bus <Byte> -IDE -LUN <Byte> -StorageDisk <StorageDisk> -VM <VM> [-BootVolume] [-JobVariable <String> ] [-PROTipID <Guid> ] [-ReturnImmediately] [-RunAsynchronously] [-StorageClassification <StorageClassification> ] [-SystemVolume] [-VirtualHardDiskFormatType {VHD | VMDK | VHDX} ] [-VMMServer <ServerConnection> ] [-VolumeType {None | Boot | System | BootAndSystem} ] [ <CommonParameters>]

Parameter Set: HostDiskToVMPParamSetSCSI

New-SCVirtualDiskDrive -Bus <Byte> -LUN <Byte> -SCSI -StorageDisk <StorageDisk> -VM <VM> [-BootVolume] [-FileName <String> ] [-JobVariable <String> ] [-Path <String> ] [-PROTipID <Guid> ] [-ReturnImmediately] [-RunAsynchronously] [-StorageClassification <StorageClassification> ] [-SystemVolume] [-VirtualHardDiskFormatType {VHD | VMDK | VHDX} ] [-VMMServer <ServerConnection> ] [-VolumeType {None | Boot | System | BootAndSystem} ] [ <CommonParameters>]

Parameter Set: newToJobGroupParamSetIDEDynamic

New-SCVirtualDiskDrive -Bus <Byte> -Dynamic -FileName <String> -IDE -JobGroup <Guid> -LUN <Byte> -VirtualHardDiskSizeMB <Int64> [-BootVolume] [-JobVariable <String> ] [-Path <String> ] [-PROTipID <Guid> ] [-ReturnImmediately] [-RunAsynchronously] [-StorageClassification

---

<StorageClassification> ] [-SystemVolume] [-VirtualHardDiskFormatType {VHD | VMDK | VHDX} ]  
[-VMMServer <ServerConnection> ] [-VolumeType {None | Boot | System | BootAndSystem} ] [  
<CommonParameters>]

Parameter Set: newToJobGroupParamSetIDEFixed

New-SCVirtualDiskDrive -Bus <Byte> -FileName <String> -Fixed -IDE -JobGroup <Guid> -LUN  
<Byte> -VirtualHardDiskSizeMB <Int64> [-BootVolume] [-JobVariable <String> ] [-Path <String> ]  
[-PROTipID <Guid> ] [-ReturnImmediately] [-RunAsynchronously] [-StorageClassification  
<StorageClassification> ] [-SystemVolume] [-VirtualHardDiskFormatType {VHD | VMDK | VHDX} ]  
[-VMMServer <ServerConnection> ] [-VolumeType {None | Boot | System | BootAndSystem} ] [  
<CommonParameters>]

Parameter Set: newToJobGroupParamSetSCSIDynamic

New-SCVirtualDiskDrive -Bus <Byte> -Dynamic -FileName <String> -JobGroup <Guid> -LUN <Byte>  
-SCSI -VirtualHardDiskSizeMB <Int64> [-BootVolume] [-JobVariable <String> ] [-Path <String> ]  
[-PROTipID <Guid> ] [-ReturnImmediately] [-RunAsynchronously] [-StorageClassification  
<StorageClassification> ] [-SystemVolume] [-VirtualHardDiskFormatType {VHD | VMDK | VHDX} ]  
[-VMMServer <ServerConnection> ] [-VolumeType {None | Boot | System | BootAndSystem} ] [  
<CommonParameters>]

Parameter Set: newToJobGroupParamSetSCSIFixed

New-SCVirtualDiskDrive -Bus <Byte> -FileName <String> -Fixed -JobGroup <Guid> -LUN <Byte> -  
SCSI -VirtualHardDiskSizeMB <Int64> [-BootVolume] [-JobVariable <String> ] [-Path <String> ]  
[-PROTipID <Guid> ] [-ReturnImmediately] [-RunAsynchronously] [-StorageClassification  
<StorageClassification> ] [-SystemVolume] [-VirtualHardDiskFormatType {VHD | VMDK | VHDX} ]  
[-VMMServer <ServerConnection> ] [-VolumeType {None | Boot | System | BootAndSystem} ] [  
<CommonParameters>]

Parameter Set: newToTemplateParamSetIDEDynamic

New-SCVirtualDiskDrive -Bus <Byte> -Dynamic -FileName <String> -IDE -LUN <Byte> -  
VirtualHardDiskSizeMB <Int64> -VMTemplate <Template> [-BootVolume] [-JobVariable <String> ]  
[-PROTipID <Guid> ] [-ReturnImmediately] [-RunAsynchronously] [-StorageClassification  
<StorageClassification> ] [-SystemVolume] [-VirtualHardDiskFormatType {VHD | VMDK | VHDX} ]  
[-VMMServer <ServerConnection> ] [-VolumeType {None | Boot | System | BootAndSystem} ] [  
<CommonParameters>]

Parameter Set: newToTemplateParamSetIDEFixed

New-SCVirtualDiskDrive -Bus <Byte> -FileName <String> -Fixed -IDE -LUN <Byte> -  
VirtualHardDiskSizeMB <Int64> -VMTemplate <Template> [-BootVolume] [-JobVariable <String> ]  
[-PROTipID <Guid> ] [-ReturnImmediately] [-RunAsynchronously] [-StorageClassification  
<StorageClassification> ] [-SystemVolume] [-VirtualHardDiskFormatType {VHD | VMDK | VHDX} ]  
[-VMMServer <ServerConnection> ] [-VolumeType {None | Boot | System | BootAndSystem} ] [  
<CommonParameters>]

Parameter Set: newToTemplateParamSetSCSIDynamic

New-SCVirtualDiskDrive -Bus <Byte> -Dynamic -FileName <String> -LUN <Byte> -SCSI -  
VirtualHardDiskSizeMB <Int64> -VMTemplate <Template> [-BootVolume] [-JobVariable <String> ]  
[-PROTipID <Guid> ] [-ReturnImmediately] [-RunAsynchronously] [-StorageClassification  
<StorageClassification> ] [-SystemVolume] [-VirtualHardDiskFormatType {VHD | VMDK | VHDX} ]  
[-VMMServer <ServerConnection> ] [-VolumeType {None | Boot | System | BootAndSystem} ] [  
<CommonParameters>]

---

Parameter Set: newToTemplateParamSetSCSIFixed

New-SCVirtualDiskDrive -Bus <Byte> -FileName <String> -Fixed -LUN <Byte> -SCSI -  
VirtualHardDiskSizeMB <Int64> -VMTemplate <Template> [-BootVolume] [-JobVariable <String> ]  
[-PROTipID <Guid> ] [-ReturnImmediately] [-RunAsynchronously] [-StorageClassification  
<StorageClassification> ] [-SystemVolume] [-VirtualHardDiskFormatType {VHD | VMDK | VHDX} ]  
[-VMMServer <ServerConnection> ] [-VolumeType {None | Boot | System | BootAndSystem} ] [  
<CommonParameters>]

Parameter Set: newToVMPParamSetIDEDynamic

New-SCVirtualDiskDrive -Bus <Byte> -Dynamic -FileName <String> -IDE -LUN <Byte> -  
VirtualHardDiskSizeMB <Int64> -VM <VM> [-BootVolume] [-JobVariable <String> ] [-Path  
<String> ] [-PROTipID <Guid> ] [-ReturnImmediately] [-RunAsynchronously] [-  
StorageClassification <StorageClassification> ] [-SystemVolume] [-VirtualHardDiskFormatType  
{VHD | VMDK | VHDX} ] [-VMMServer <ServerConnection> ] [-VolumeType {None | Boot | System |  
BootAndSystem} ] [ <CommonParameters>]

Parameter Set: newToVMPParamSetIDEFixed

New-SCVirtualDiskDrive -Bus <Byte> -FileName <String> -Fixed -IDE -LUN <Byte> -  
VirtualHardDiskSizeMB <Int64> -VM <VM> [-BootVolume] [-JobVariable <String> ] [-Path  
<String> ] [-PROTipID <Guid> ] [-ReturnImmediately] [-RunAsynchronously] [-  
StorageClassification <StorageClassification> ] [-SystemVolume] [-VirtualHardDiskFormatType  
{VHD | VMDK | VHDX} ] [-VMMServer <ServerConnection> ] [-VolumeType {None | Boot | System |  
BootAndSystem} ] [ <CommonParameters>]

Parameter Set: newToVMPParamSetSCSIDynamic

New-SCVirtualDiskDrive -Bus <Byte> -Dynamic -FileName <String> -LUN <Byte> -SCSI -  
VirtualHardDiskSizeMB <Int64> -VM <VM> [-BootVolume] [-JobVariable <String> ] [-Path  
<String> ] [-PROTipID <Guid> ] [-ReturnImmediately] [-RunAsynchronously] [-  
StorageClassification <StorageClassification> ] [-SystemVolume] [-VirtualHardDiskFormatType  
{VHD | VMDK | VHDX} ] [-VMMServer <ServerConnection> ] [-VolumeType {None | Boot | System |  
BootAndSystem} ] [ <CommonParameters>]

Parameter Set: newToVMPParamSetSCSIFixed

New-SCVirtualDiskDrive -Bus <Byte> -FileName <String> -Fixed -LUN <Byte> -SCSI -  
VirtualHardDiskSizeMB <Int64> -VM <VM> [-BootVolume] [-JobVariable <String> ] [-Path  
<String> ] [-PROTipID <Guid> ] [-ReturnImmediately] [-RunAsynchronously] [-  
StorageClassification <StorageClassification> ] [-SystemVolume] [-VirtualHardDiskFormatType  
{VHD | VMDK | VHDX} ] [-VMMServer <ServerConnection> ] [-VolumeType {None | Boot | System |  
BootAndSystem} ] [ <CommonParameters>]

Parameter Set: newVirtualDiskDriveDifferencing

New-SCVirtualDiskDrive -Bus <Byte> -Differencing -FileName <String> -LUN <Byte> -  
ParentVirtualHardDiskPath <String> [-BootVolume] [-IDE] [-JobGroup <Guid> ] [-JobVariable  
<String> ] [-Path <String> ] [-PROTipID <Guid> ] [-ReturnImmediately] [-RunAsynchronously]  
[-SCSI] [-StorageClassification <StorageClassification> ] [-SystemVolume] [-  
VirtualHardDiskFormatType {VHD | VMDK | VHDX} ] [-VM <VM> ] [-VMMServer <ServerConnection> ]  
[-VolumeType {None | Boot | System | BootAndSystem} ] [ <CommonParameters>]

---

## Detailed Description

The New-SCVirtualDiskDrive cmdlet creates a virtual disk drive object on a virtual machine deployed on a host managed by Virtual Machine Manager (VMM), or creates a virtual disk drive object on a template in the VMM library.

With System Center 2012 Service Pack 1 (SP1), you can now create a differencing disk by using the Differencing and ParentVirtualHardDiskPath parameters. A differencing disk is a disk associated in a parent-child relationship to another disk that you want to leave in tact. You can make changes to the data or operating system on the child disk without affecting the parent disk, so that you can revert changes to the child disk. The child disk must have the same virtual hard disk format as the parent disk (either VHD or VHDX), which you can set with the VirtualHardDiskFormat parameter. See example 6 for a demonstration of how to create a differencing disk and attach it to an existing virtual machine.

A virtual hard disk file (a Windows-based .vhd or .vhdx file or a VMware-based.vmdk file) that is stored on a VMM library share, but is not attached to a virtual disk drive, exists as a standalone object in the library.

A pass-through disk is a disk on a Hyper-V or VMware ESX host that a virtual machine on that host can use as an alternative to using a virtual hard disk. The corresponding term used by VMware for a pass-through disk is Raw Device Mapping, or RDM. The host disk can be either a hard disk on the host or a logical unit on a Storage Area Network (SAN). VMM lets the virtual machine bypass the host's file system and access the pass-through disk directly.

### TYPE OF HOSTPASS-THROUGH DISK SUPPORT

-----

Hyper-VSupports pass-through disks

Hyper-VSupports converting a pass-through disk to a VHD

VMware ESXSupports pass-through disks (RDP), but not disk conversion

Citrix XenServerDoes not support pass-through disks

NOTE: You cannot create a checkpoint of a pass-through disk because checkpoint creation is designed to work with virtual hard disks.

## Parameters

### -AnyStorageDisk

Identifies a placeholder parameter that is used to indicate the creation of pass-through disks within a new virtual machine job group.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false

---

Accept Wildcard Characters?	false
-----------------------------	-------

## **-BootVolume**

Indicates that the volume attached to the VirtualDiskDrive is a boot volume.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Bus<Byte>**

Specifies the IDE bus to which to attach a virtual disk drive or virtual DVD drive, or the SCSI bus to which to attach a virtual disk drive.

Example format: -IDE -Bus 1 -LUN 0

Example format: -SCSI -Bus 0 -LUN 1

NOTE: See the examples for a specific cmdlet to determine how that cmdlet uses this parameter.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Differencing**

Specifies that the virtual hard disk should be created as a differencing disk.

Aliases	none
Required?	true

---

Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Dynamic**

Specifies that a virtual hard disk can expand dynamically.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-FileName<String>**

Specifies the file name to use when you rename a virtual hard disk file as you add it to a virtual machine.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Fixed**

Specifies that a virtual hard disk is fixed in size.

Aliases	none
---------	------

Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-IDE**

Specifies IDE as the bus type to which to attach a virtual disk drive object or a virtual DVD drive object configured on a virtual machine or on a template.

Example format: -IDE -Bus 0 -LUN 1

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-JobGroup<Guid>**

Specifies an identifier for a series of commands that will run as a set just before the final command that includes the same job group identifier runs.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.



Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-LUN<Byte>**

Specifies the logical unit number (LUN) for a virtual disk drive object or for a virtual DVD drive object on an IDE bus, or for a virtual disk drive object on a SCSI bus.

Example format: -IDE -Bus 1 -LUN 0

Example format: -SCSI -Bus 0 -LUN 1

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-ParentVirtualHardDiskPath<String>**

Specifies the path to the parent virtual hard disk to be used when creating a differencing disk.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-Path<String>**

Specifies the destination path for the operation.

Example formats:

Local path -Path "F:\"

UNC path -Path "\\Library\Templates"

Volume GUID path -Path "\\?\Volume{4703c1ea-8ae7-11db-b473-00123f7603e3}\\"

VMware ESX path -Path "[storage1]\MyVMwareFolderForVMs\MyVM.vmx"

Citrix XenServer path - Path "Local storage[99b6212f-b63d-c676-25f9-d6c460992de7]"

Wildcards are supported for "Get" cmdlets and when you specify the UNC path:

Example format:

UNC path -Path "\\VMHostServer\MyVMs\\*VM\*"

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-ReturnImmediately**

Indicates that control is returned to the calling process immediately, before the job is created.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-SCSI**

Specifies SCSI as the bus type to which to attach a virtual disk drive object configured on a virtual machine or on a template.

Example format: -SCSI -Bus 0 -LUN 0

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-StorageClassification<StorageClassification>**

Specifies a storage classification object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-StorageDisk<StorageDisk>**

Specifies a disk on a Hyper-V or VMware ESX host that a virtual machine on that host can use instead of using a virtual hard disk. This disk is referred to as a pass-through disk (the corresponding VMware term is Raw Device Mapping, or RDM). The host disk is either a local hard disk or a logical unit on a Storage Area Network (SAN). VMM lets the virtual machine bypass the host's file system and access the pass-through disk directly.

TYPE OF HOST    PASS-THROUGH DISK SUPPORT

-----    -----

Hyper-V       Supports pass-through disks

Supports converting a pass-through disk to a VHD

VMware ESX   Supports pass-through disks (RDP), but not disk conversion

Citrix XenServer Does not support pass-through disks

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-SystemVolume**

Indicates that the volume attached to the VirtualDiskDrive is a system volume.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-UseLocalVirtualHardDisk**

Verifies that the VHD file (or files) to be used to create the virtual machine exist and are stored on the destination host.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VirtualHardDisk<StandaloneVirtualHardDisk>**

Specifies a virtual hard disk object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VirtualHardDiskFormatType<VHDFormatType>**

Specifies the hard disk format type. Valid values are VHD, VHDX, and VMDK.

---

The acceptable values for this parameter are:

VHD	
VMDK	
VHDX	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VirtualHardDiskSizeMB<Int64>**

Specifies, in megabytes (MB), the size of a fixed virtual hard disk file or the maximum possible size of a dynamically expanding virtual hard disk file.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-VM<VM>**

Specifies a virtual machine object.

Aliases	none
Required?	true
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-VMTemplate<Template>**

Specifies a VMM template object used to create virtual machines.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-VolumeType<VolumeType>**

Specifies the volume type for a virtual hard disk. Valid values: Boot, System, BootAndSystem, None.

The acceptable values for this parameter are:

None	
Boot	

System	
BootAndSystem	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **VirtualDiskDrive**

## Notes

- Requires a VMM virtual hard disk object, which can be retrieved by using the Get-SCVirtualHardDisk cmdlet.

## Examples

### 1: Create a virtual disk drive on a template and attach an existing virtual hard disk to the virtual disk drive.

The first command gets the virtual hard disk object named "Blank Disk - Small" from the VMM library stores the object in the \$VHD variable.

The second command gets the virtual machine template object named VMTemplate01 from the library stores the in the \$Template variable.

The last command creates a new virtual disk drive on VMTemplate01 and attaches the virtual hard disk stored in \$VHD to the second channel in the second slot of the IDE bus on the virtual disk drive.

```
PS C:\> $VHD = Get-SCVirtualHardDisk -Name "Blank Disk - Small.vhd"
```

```
PS C:\> $VMTemplate = Get-SCVMTemplate | where {$_.Name -eq "VMTemplate01"}
```



---

```
PS C:\> New-SCVirtualDiskDrive -VMTemplate $VMTemplate -IDE -Bus 1 -Lun 1 -VirtualHardDisk $VHD
```

## 2: Create a virtual disk drive and add it to an existing virtual machine.

The first command gets the virtual machine object named VM01 and stores the object in the \$VM variable.

The second command creates a new dynamic virtual disk drive on the first IDE channel in the second slot of the virtual machine and specifies its size as 19.5 GB.

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM01"
```

```
PS C:\> New-SCVirtualDiskDrive -VM $VM -Dynamic -Filename "Test" -IDE -Size 20000 -Bus 0 -LUN 1
```

## 3: Create a virtual disk drive from an existing VHD and attach it to a new virtual machine.

The first command generates a new globally unique identifier (GUID) and stores the GUID string in variable \$JobGroupID. The job group ID functions as an identifier that groups subsequent commands that include \$JobGroupID into a single job group.

The second command gets the virtual hard disk object from the VMM library location \\LibraryServer01.Contoso.com\MSSCVMMLibrary\VHDs\Blank Disk - Large.vhd and stores the object in the \$VHD variable.

The third command creates a new virtual hard disk drive object and assigns the new object to IDE Bus 0 and LUN 1. This command also attaches the virtual hard disk stored in \$VHD to the new object. By using the JobGroup parameter, this command will run just before the last command that invokes the JobGroup and associate the new virtual hard drive object with the new virtual machine created in the last command.

The fourth command gets the hardware profile object that contains the string "NewHWProfile01" in its name and stores the object in the \$HWProfile variable.

The fifth command gets the host object named VMHost03, stores the host object in the \$VMHost variable.

The last command creates a new virtual machine named VM10 using the hardware settings stored in \$HWProfile, deploys the virtual machine on VMHost03, and specifies that the virtual machine is not automatically started when the host starts and is put into a saved state when the virtualization service stops. This command uses the JobGroup parameter to indicate that any previous cmdlets that use the same JobGroup ID is run prior to creating the new virtual machine. In this case, the New-SCVirtualDiskDrive cmdlet from the third command creates a virtual disk drive and associates it with the new virtual machine.

```
PS C:\> $JobGroupID = [Guid]::NewGuid().ToString()
```

```
PS C:\> $VHD = Get-SCVirtualHardDisk -VMMServer "VMMServer01.Contoso.com" | where {$_.Location -eq "\\LibServer01.Contoso.com\MSSCVMMLibrary\VHDs\Blank Disk - Large.vhd"}
```

---

```
PS C:\> New-SCVirtualDiskDrive -IDE -Bus 0 -LUN 1 -JobGroup $JobGroupID -VirtualHardDisk $VHD
PS C:\> $HWProfile = Get-SCHardwareProfile | where {$_.Name -match "NewHWProfile01"}
PS C:\> $VMHost = Get-SCVMHost -ComputerName "VMHost03"
PS C:\> New-SCVirtualMachine -Name "VM10" -Description "New Virtual Machine VM10" -VMMServer "VMMServer01.Contoso.com" -Owner "Contoso\Katarina" -VMHost $VMHost -Path "D:\VirtualMachinePath" -HardwareProfile $HWProfile -JobGroup $JobGroupID -RunAsynchronously -StartAction NeverAutoTurnOnVM -StopAction SaveVM
```

## 4. Create a virtual disk drive using a host disk and attach it to an existing virtual machine.

The first command gets the virtual machine object named VM04 and stores the object in the \$VM variable.

The second command gets the host object named VMHost04 and stores the object in the \$VMHost variable.

The third command gets all storage disk objects on VMHost04 that are pass-through capable and stores the objects in the \$HostDisk variable. Using the "@" symbol and parentheses ensures that the command stores the results in an array, in case the command returns a single object or a null value.

The last command creates a new virtual disk drive object that is connected to a physical host disk on VMHost04. The virtual disk drive is attached to the second slot of the first SCSI bus on VM04. This example assumes the virtual machine already has a SCSI controller.

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM04"
PS C:\> $VMHost = Get-SCVMHost -ComputerName "VMHost04.Contoso.com"
PS C:\> $HostDisk = @(Get-SCStorageDisk -VMHost $VMHost | where {$_.IsPassThroughCapable -eq $True})
PS C:\> New-SCVirtualDiskDrive -VM $VM -HostDisk $HostDisk[0] -SCSI -Bus 0 -LUN 1
```

## 5: Create a virtual disk drive from an existing VHDX and attach it to an existing virtual machine.

The first command gets the virtual hard disk object named Blank Disk - Large.vhdx from the VMM library and stores the object in the \$VHDX variable.

The second command gets the virtual machine object named VM05 and stores the object in the \$VM variable.

The last command creates a virtual disk drive from Blank Disk - Large.vhdx and attaches the new virtual disk drive to VM05.

```
PS C:\> $VHDX = Get-SCVirtualHardDisk -VMMServer "VMMServer01.Contoso.com" | where {$_.Location -eq "\\LibServer01.Contoso.com\MSSCMMLibrary\VHDXs\Blank Disk - Large.vhdx"}
PS C:\> $VM = Get-SCVirtualMachine -Name "VM05"
PS C:\> New-SCVirtualDiskDrive -VM $VM -VirtualHardDisk $VHDX -IDE -Bus 0 -LUN 1
```

---

## 6: Create a differencing disk and attach it to an existing virtual machine.

The first command gets the virtual machine object named VM06 and stores the object in the \$VM variable.

The second command creates a differencing disk named DiffDisk.vhdx with the specified parent disk, and attaches the differencing disk to VM06.

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM06"
```

```
PS C:\> New-SCVirtualDiskDrive -Bus 1 -LUN 1 -FileName "DiffDisk" -ParentVirtualHardDiskPath "C:\VirtualMachines\ParentDisk01.vhdx" -Differencing -VirtualHardDiskFormat "VHDX" -IDE -VM $VM
```

### Related topics

[Compress-SCVirtualDiskDrive](#)

[Convert-SCVirtualDiskDrive](#)

[Expand-SCVirtualDiskDrive](#)

[Get-SCVirtualDiskDrive](#)

[Remove-SCVirtualDiskDrive](#)

[Set-SCVirtualDiskDrive](#)

[Get-SCVirtualHardDisk](#)

[Move-SCVirtualHardDisk](#)

[Remove-SCVirtualHardDisk](#)

[Set-SCVirtualHardDisk](#)

---

# New-SCVirtualDVDDrive

---

## New-SCVirtualDVDDrive

Creates a virtual DVD drive on a virtual machine, a virtual machine template, or a hardware profile used in VMM.

### Syntax

Parameter Set: HardwareProfile

```
New-SCVirtualDVDDrive -Bus <Byte> -HardwareProfile <HardwareProfile> -LUN <Byte> [-AnyVMHostDrive] [-ISO <ISO> ] [-JobVariable <String> ] [-Link] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMHostDrive <String> ] [ <CommonParameters>]
```

Parameter Set: JobGroup

```
New-SCVirtualDVDDrive -Bus <Byte> -JobGroup <Guid> -LUN <Byte> [-AnyVMHostDrive] [-ISO <ISO> ] [-JobVariable <String> ] [-Link] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMHostDrive <String> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: Template

```
New-SCVirtualDVDDrive -Bus <Byte> -LUN <Byte> -VMTemplate <Template> [-AnyVMHostDrive] [-ISO <ISO> ] [-JobVariable <String> ] [-Link] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMHostDrive <String> ] [ <CommonParameters>]
```

Parameter Set: VM

```
New-SCVirtualDVDDrive -Bus <Byte> -LUN <Byte> -VM <VM> [-AnyVMHostDrive] [-ISO <ISO> ] [-JobVariable <String> ] [-Link] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMHostDrive <String> ] [ <CommonParameters>]
```

### Detailed Description

The New-SCVirtualDVDDrive cmdlet creates a virtual DVD drive object on a virtual machine, a virtual machine template, or a hardware profile used in a Virtual Machine Manager (VMM) environment. By default, the virtual DVD drive created by New-SCVirtualDVDDrive is not connected to any media. You can use the Set-SCVirtualDVDDrive cmdlet to connect a virtual DVD drive to a physical DVD drive on a virtual machine host or to an ISO image.

NOTE: You can connect a virtual DVD drive to an IDE device on a virtual machine but you cannot connect a virtual DVD drive to a SCSI adapter on a virtual machine.

---

## Parameters

### **-AnyVMHostDrive**

Indicates that a virtual DVD or floppy drive on a virtual machine will be connected to any corresponding physical drive on a host. This mapping occurs when you deploy a stored virtual machine on a host, or when you use a template or hardware profile to create and deploy a virtual machine on a host.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Bus<Byte>**

Specifies the IDE bus to which to attach a virtual disk drive or virtual DVD drive, or the SCSI bus to which to attach a virtual disk drive.

Example format: -IDE -Bus 1 -LUN 0

Example format: -SCSI -Bus 0 -LUN 1

NOTE: See the examples for a specific cmdlet to determine how that cmdlet uses this parameter.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-HardwareProfile<HardwareProfile>**

Specifies a hardware profile object.

Aliases	none
---------	------

Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-ISO<ISO>**

Specifies an ISO object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-JobGroup<Guid>**

Specifies an identifier for a series of commands that will run as a set just before the final command that includes the same job group identifier runs.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Link**

Indicates that a resource should be linked to instead of copied.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-LUN<Byte>**

Specifies the logical unit number (LUN) for a virtual disk drive object or for a virtual DVD drive object on an IDE bus, or for a virtual disk drive object on a SCSI bus.

Example format: -IDE -Bus 1 -LUN 0

Example format: -SCSI -Bus 0 -LUN 1

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VM<VM>**

Specifies a virtual machine object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false



---

## **-VMHostDrive<String>**

Specifies a drive on a virtual machine host.

Example formats:

Hyper-V host hard drive: "C:"

Hyper-V host floppy drive: "A:"

VMware ESX host hard drive: "/dev/tools"

VMware ESX host floppy drive: "/dev/sda"

Citrix XenServer host hard drive: "Local storage[99b6212f-b63d-c676-25f9-d6c460992de7]"

Citrix XenServer host floppy drive: Not Supported

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **-VMTemplate<Template>**

Specifies a VMM template object used to create virtual machines.

Aliases	none
Required?	true

---

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **VirtualDVDDrive**

## Notes

- Requires a VMM virtual machine object, virtual machine template object, or hardware profile object. You can retrieve these objects by using the Get-SCVirtualMachine, Get-SCVMTemplate, or Get-SCHardwareProfile cmdlets, respectively.

## Examples

### 1: Create a virtual DVD drive on a virtual machine.

The first command gets the virtual machine object named VM01 and stores the object in the \$VM variable.

The second command creates a virtual DVD drive on VM01 and attaches the virtual DVD drive to Secondary channel (1) by specifying IDE Bus 1 and LUN 1.

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM01"
PS C:\> New-SCVirtualDVDDrive -VM $VM -Bus 1 -LUN 1
```

### 2: Create a virtual DVD drive on a virtual machine template.

The first command gets the virtual machine template object named VMTemplate01 and stores the object in the \$Template variable.

The second command creates a virtual DVD drive on VMTemplate01 that attaches a virtual DVD drive to Secondary Channel (1) on the IDE bus when the template is used to create a virtual machine.

```
PS C:\> $VMTemplate = Get-SCVMTemplate | where { $_.Name -eq "VMTemplate01" }
PS C:\> New-SCVirtualDVDDrive -VMTemplate $VMTemplate -Bus 1 -LUN 1
```

---

### 3: Create a virtual DVD drive on a hardware profile.

The first command gets the hardware profile object named NewHardwareProfile01 and stores the object in the \$HWProfile variable.

The second command creates a virtual DVD drive on HardwareProfile1 that attaches a virtual DVD drive to Secondary Channel (1) on the IDE bus when the hardware profile is used to create a virtual machine,

```
PS C:\> $HWProfile = Get-SCHardwareProfile | where { $_.Name -eq "NewHWProfile01" }
PS C:\> New-SCVirtualDVDDrive -HardwareProfile $HWProfile -Bus 1 -LUN 1
```

### 4. Create a virtual machine with a virtual DVD drive that connects to any available physical DVD drive on the host.

The first command creates a new GUID string and stores it to variable \$JobGroupId. This GUID is a job group ID that functions as an identifier that groups subsequent commands that include this identifier into a single job group.

The second command creates a new virtual DVD drive object and specifies that this new virtual DVD drive can use any available physical DVD drive. The command will attach the new virtual DVD drive to the first slot of the second IDE channel (IDE is the only bus type that a virtual DVD drive can be attached to). Using the job group ID specifies that that this command will not run until just before the final command that includes the JobGroup parameter runs.

The third command gets the host object named VMHost04 and stores the object in the \$VMHost variable.

The last command creates a virtual machine, names it VM04, provides a description, assigns an owner, and specifies the location on the host to store the virtual machine. The command uses the job group ID to run the New-SCVirtualDVDDrive command just before the New-SCVirtualMachine command runs; the resulting virtual DVD drive object is associated with the new virtual machine.

```
PS C:\> $JobGroupId = [Guid]::NewGuid().ToString()
PS C:\> New-SCVirtualDVDDrive -VMMServer "VMMServer01.Contoso.com" -JobGroup $JobGroupId -
Bus 1 -LUN 0 -AnyVMHostDrive
PS C:\> $VMHost = Get-SCVMHost -ComputerName "VMHost04"
PS C:\> New-SCVirtualMachine -Name "VM04" -Description "A new VM with a DVD drive" -
VMMServer "VMMServer01.Contoso.com" -Owner "Contoso\Katarina" -VMHost $VMHost -Path
"D:\VirtualMachinePath" -StartVM -JobGroup $JobGroupId
```

### 5. Add a new virtual DVD drive to an existing virtual machine and attach an ISO file from the library to the drive.

The first command gets the virtual machine object named VM05 and stores the object in the \$VM variable.

---

The second command gets the ISO object named WindowsServer2008R2.iso and stores the object in the \$ISO variable.

The last command creates a new virtual DVD drive on VM05, attaches it to the specified location on the IDE bus, and links it with the ISO image stored in \$ISO.

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM05"
```

```
PS C:\> $ISO = Get-SCISO | where {$_.Name -eq "WindowsServer2008R2.iso"}
```

```
PS C:\> New-SCVirtualDVDDrive -VM $VM -ISO $ISO -Bus 1 -LUN 1
```

## Related topics

[Get-SCHardwareProfile](#)

[Get-SCVirtualMachine](#)

[Get-SCVMMServer](#)

[Get-SCVMTemplate](#)

[Remove-SCVirtualDVDDrive](#)

[Set-SCVirtualDVDDrive](#)

---

# New-SCVirtualMachine

---

## New-SCVirtualMachine

Creates a virtual machine to be managed by VMM.

### Syntax

Parameter Set: NewStoredVmFromHardwareProfile

```
New-SCVirtualMachine [-Name] <String> -LibraryServer <LibraryServer> -SharePath <String> [-CPUCount <Byte> ] [-CPUCyclesLimitPercent <UInt16> ] [-CPUCyclesReservePercent <UInt16> ] [-CPULimitForMigration <Boolean> ] [-CPULimitFunctionality <Boolean> ] [-CPUPerVirtualNumaNodeMaximum <Byte> ] [-CPURelativeWeight <Int32> ] [-CPUType <ProcessorType> ] [-DelayStartSeconds <Int32> ] [-Description <String> ] [-DynamicMemoryBufferPercentage <Int32> ] [-DynamicMemoryEnabled <Boolean> ] [-DynamicMemoryMaximumMB <Int32> ] [-DynamicMemoryMinimumMB <UInt16> ] [-HardwareProfile <HardwareProfile> ] [-JobGroup <Guid> ] [-JobVariable <String> ] [-LinuxAdministratorSSHKey <SSHKey> ] [-LinuxDomainName <String> ] [-MemoryMB <Int32> ] [-MemoryPerVirtualNumaNodeMaximumMB <Int32> ] [-MemoryWeight <Int32> ] [-MonitorMaximumCount <Int32> ] [-MonitorMaximumResolution <String> ] [-NumaIsolationRequired <Boolean> ] [-OperatingSystem <OperatingSystem> ] [-Owner <String> ] [-PROTipID <Guid> ] [-ReturnImmediately] [-RunAsynchronously] [-SkipInstallVirtualizationGuestServices] [-StartAction {NeverAutoTurnOnVM | AlwaysAutoTurnOnVM | TurnOnVMIfRunningWhenVSSStopped} ] [-StopAction {SaveVM | TurnOffVM | ShutdownGuestOS} ] [-UseLocalVirtualHardDisk] [-UserRole <UserRole> ] [-VirtualNumaNodesPerSocketMaximum <Byte> ] [-VirtualVideoAdapterEnabled <Boolean> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: NewStoredVmFromVirtualDisk

```
New-SCVirtualMachine [-Name] <String> -LibraryServer <LibraryServer> -SharePath <String> -VirtualHardDisk <StandaloneVirtualHardDisk> [-CPUCount <Byte> ] [-CPUCyclesLimitPercent <UInt16> ] [-CPUCyclesReservePercent <UInt16> ] [-CPULimitForMigration <Boolean> ] [-CPULimitFunctionality <Boolean> ] [-CPUPerVirtualNumaNodeMaximum <Byte> ] [-CPURelativeWeight <Int32> ] [-CPUType <ProcessorType> ] [-DelayStartSeconds <Int32> ] [-Description <String> ] [-DynamicMemoryBufferPercentage <Int32> ] [-DynamicMemoryEnabled <Boolean> ] [-DynamicMemoryMaximumMB <Int32> ] [-DynamicMemoryMinimumMB <UInt16> ] [-HardwareProfile <HardwareProfile> ] [-JobGroup <Guid> ] [-JobVariable <String> ] [-LinuxAdministratorSSHKey <SSHKey> ] [-LinuxDomainName <String> ] [-MemoryMB <Int32> ] [-MemoryPerVirtualNumaNodeMaximumMB <Int32> ] [-MemoryWeight <Int32> ] [-MonitorMaximumCount <Int32> ] [-MonitorMaximumResolution <String> ] [-NumaIsolationRequired <Boolean> ] [-OperatingSystem <OperatingSystem> ] [-Owner <String> ] [-PROTipID <Guid> ] [-ReturnImmediately] [-RunAsynchronously] [-SkipInstallVirtualizationGuestServices] [-StartAction {NeverAutoTurnOnVM | AlwaysAutoTurnOnVM | TurnOnVMIfRunningWhenVSSStopped} ] [-StopAction {SaveVM | TurnOffVM | ShutdownGuestOS} ] [-UseLocalVirtualHardDisk] [-UserRole <UserRole> ] [-VirtualNumaNodesPerSocketMaximum <Byte> ] [-VirtualVideoAdapterEnabled <Boolean> ] [ <CommonParameters>]
```

Parameter Set: NewStoredVmFromVm

```
New-SCVirtualMachine [-Name] <String> -LibraryServer <LibraryServer> -SharePath <String> -VM
```

---

<VM> [-CPUCount <Byte> ] [-CPUCyclesLimitPercent <UInt16> ] [-CPUCyclesReservePercent <UInt16> ] [-CPULimitForMigration <Boolean> ] [-CPULimitFunctionality <Boolean> ] [-CPUPerVirtualNumaNodeMaximum <Byte> ] [-CPURelativeWeight <Int32> ] [-CPUType <ProcessorType> ] [-DelayStartSeconds <Int32> ] [-Description <String> ] [-DynamicMemoryBufferPercentage <Int32> ] [-DynamicMemoryEnabled <Boolean> ] [-DynamicMemoryMaximumMB <Int32> ] [-DynamicMemoryMinimumMB <UInt16> ] [-HardwareProfile <HardwareProfile> ] [-JobGroup <Guid> ] [-JobVariable <String> ] [-LinuxAdministratorSSHKey <SSHKey> ] [-LinuxDomainName <String> ] [-MemoryMB <Int32> ] [-MemoryPerVirtualNumaNodeMaximumMB <Int32> ] [-MemoryWeight <Int32> ] [-MonitorMaximumCount <Int32> ] [-MonitorMaximumResolution <String> ] [-NumaIsolationRequired <Boolean> ] [-OperatingSystem <OperatingSystem> ] [-Owner <String> ] [-PROTipID <Guid> ] [-ReturnImmediately] [-RunAsynchronously] [-SkipInstallVirtualizationGuestServices] [-StartAction {NeverAutoTurnOnVM | AlwaysAutoTurnOnVM | TurnOnVMIfRunningWhenVSSStopped} ] [-StopAction {SaveVM | TurnOffVM | ShutdownGuestOS} ] [-UseLocalVirtualHardDisk] [-UserRole <UserRole> ] [-VirtualNumaNodesPerSocketMaximum <Byte> ] [-VirtualVideoAdapterEnabled <Boolean> ] [ <CommonParameters>]

Parameter Set: NewVmFromComputerTierScaleOut

New-SCVirtualMachine [-Name] <String> -ComputerTier <ComputerTier> [-ComputerName <String> ] [-CPUCount <Byte> ] [-CPUCyclesLimitPercent <UInt16> ] [-CPUCyclesReservePercent <UInt16> ] [-CPULimitForMigration <Boolean> ] [-CPULimitFunctionality <Boolean> ] [-CPUPerVirtualNumaNodeMaximum <Byte> ] [-CPURelativeWeight <Int32> ] [-CPUType <ProcessorType> ] [-DelayStartSeconds <Int32> ] [-Description <String> ] [-DynamicMemoryBufferPercentage <Int32> ] [-DynamicMemoryEnabled <Boolean> ] [-DynamicMemoryMaximumMB <Int32> ] [-DynamicMemoryMinimumMB <UInt16> ] [-HardwareProfile <HardwareProfile> ] [-JobGroup <Guid> ] [-JobVariable <String> ] [-LinuxAdministratorSSHKey <SSHKey> ] [-LinuxDomainName <String> ] [-MemoryMB <Int32> ] [-MemoryPerVirtualNumaNodeMaximumMB <Int32> ] [-MemoryWeight <Int32> ] [-MonitorMaximumCount <Int32> ] [-MonitorMaximumResolution <String> ] [-NumaIsolationRequired <Boolean> ] [-OperatingSystem <OperatingSystem> ] [-Owner <String> ] [-PROTipID <Guid> ] [-ReturnImmediately] [-RunAsynchronously] [-SkipInstallVirtualizationGuestServices] [-StartAction {NeverAutoTurnOnVM | AlwaysAutoTurnOnVM | TurnOnVMIfRunningWhenVSSStopped} ] [-StopAction {SaveVM | TurnOffVM | ShutdownGuestOS} ] [-UseLocalVirtualHardDisk] [-UserRole <UserRole> ] [-VirtualNumaNodesPerSocketMaximum <Byte> ] [-VirtualVideoAdapterEnabled <Boolean> ] [ <CommonParameters>]

Parameter Set: NewVmFromHWProfile

New-SCVirtualMachine [-Name] <String> -Path <String> -VMHost <Host> [-BlockDynamicOptimization <Boolean> ] [-CPUCount <Byte> ] [-CPUCyclesLimitPercent <UInt16> ] [-CPUCyclesReservePercent <UInt16> ] [-CPULimitForMigration <Boolean> ] [-CPULimitFunctionality <Boolean> ] [-CPUPerVirtualNumaNodeMaximum <Byte> ] [-CPURelativeWeight <Int32> ] [-CPUType <ProcessorType> ] [-DelayStartSeconds <Int32> ] [-Description <String> ] [-DRProtectionRequired <Boolean> ] [-DynamicMemoryBufferPercentage <Int32> ] [-DynamicMemoryEnabled <Boolean> ] [-DynamicMemoryMaximumMB <Int32> ] [-DynamicMemoryMinimumMB <UInt16> ] [-HardwareProfile <HardwareProfile> ] [-HighlyAvailable <Boolean> ] [-JobGroup <Guid> ] [-JobVariable <String> ] [-LinuxAdministratorSSHKey <SSHKey> ] [-LinuxDomainName <String> ] [-MemoryMB <Int32> ] [-MemoryPerVirtualNumaNodeMaximumMB <Int32> ] [-MemoryWeight <Int32> ] [-MonitorMaximumCount <Int32> ] [-MonitorMaximumResolution <String> ] [-NumaIsolationRequired <Boolean> ] [-OperatingSystem <OperatingSystem> ] [-Owner <String> ] [-PROTipID <Guid> ] [-ReturnImmediately] [-RunAsynchronously] [-SkipInstallVirtualizationGuestServices] [-StartAction {NeverAutoTurnOnVM | AlwaysAutoTurnOnVM | TurnOnVMIfRunningWhenVSSStopped} ] [-StartVM] [-

---

StopAction {SaveVM | TurnOffVM | ShutdownGuestOS} ] [-UseLocalVirtualHardDisk] [-UserRole <UserRole> ] [-VirtualNumaNodesPerSocketMaximum <Byte> ] [-VirtualVideoAdapterEnabled <Boolean> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]

Parameter Set: NewVmFromTemplate

New-SCVirtualMachine [-Name] <String> -Path <String> -VMHost <Host> -VMTemplate <Template> [-AnswerFile <Script> ] [-BlockDynamicOptimization <Boolean> ] [-ComputerName <String> ] [-CPUCount <Byte> ] [-CPUCyclesLimitPercent <UInt16> ] [-CPUCyclesReservePercent <UInt16> ] [-CPULimitForMigration <Boolean> ] [-CPULimitFunctionality <Boolean> ] [-CPUPerVirtualNumaNodeMaximum <Byte> ] [-CPURelativeWeight <Int32> ] [-CPUType <ProcessorType> ] [-DelayStartSeconds <Int32> ] [-Description <String> ] [-Domain <String> ] [-DomainJoinCredential <VMCrediental> ] [-DRProtectionRequired <Boolean> ] [-DynamicMemoryBufferPercentage <Int32> ] [-DynamicMemoryEnabled <Boolean> ] [-DynamicMemoryMaximumMB <Int32> ] [-DynamicMemoryMinimumMB <UInt16> ] [-FullName <String> ] [-GuestOSProfile <GuestOSProfile> ] [-GuiRunOnceCommands <String[]> ] [-HardwareProfile <HardwareProfile> ] [-HighlyAvailable <Boolean> ] [-JobGroup <Guid> ] [-JobVariable <String> ] [-LinuxAdministratorSSHKey <SSHKey> ] [-LinuxDomainName <String> ] [-LocalAdministratorCredential <VMCrediental> ] [-MemoryMB <Int32> ] [-MemoryPerVirtualNumaNodeMaximumMB <Int32> ] [-MemoryWeight <Int32> ] [-MergeAnswerFile <Boolean> ] [-MonitorMaximumCount <Int32> ] [-MonitorMaximumResolution <String> ] [-NumaIsolationRequired <Boolean> ] [-OperatingSystem <OperatingSystem> ] [-OrganizationName <String> ] [-Owner <String> ] [-ProductKey <String> ] [-PROTipID <Guid> ] [-ReturnImmediately] [-RunAsynchronously] [-SkipInstallVirtualizationGuestServices] [-StartAction {NeverAutoTurnOnVM | AlwaysAutoTurnOnVM | TurnOnVMIfRunningWhenVSStopped} ] [-StartVM] [-StopAction {SaveVM | TurnOffVM | ShutdownGuestOS} ] [-TimeZone <Int32> ] [-UseLocalVirtualHardDisk] [-UserRole <UserRole> ] [-VirtualNumaNodesPerSocketMaximum <Byte> ] [-VirtualVideoAdapterEnabled <Boolean> ] [-Workgroup <String> ] [ <CommonParameters>]

Parameter Set: NewVmFromVirtualDisk

New-SCVirtualMachine [-Name] <String> -Path <String> -VirtualHardDisk <StandaloneVirtualHardDisk> -VMHost <Host> [-BlockDynamicOptimization <Boolean> ] [-CPUCount <Byte> ] [-CPUCyclesLimitPercent <UInt16> ] [-CPUCyclesReservePercent <UInt16> ] [-CPULimitForMigration <Boolean> ] [-CPULimitFunctionality <Boolean> ] [-CPUPerVirtualNumaNodeMaximum <Byte> ] [-CPURelativeWeight <Int32> ] [-CPUType <ProcessorType> ] [-DelayStartSeconds <Int32> ] [-Description <String> ] [-DRProtectionRequired <Boolean> ] [-DynamicMemoryBufferPercentage <Int32> ] [-DynamicMemoryEnabled <Boolean> ] [-DynamicMemoryMaximumMB <Int32> ] [-DynamicMemoryMinimumMB <UInt16> ] [-HardwareProfile <HardwareProfile> ] [-HighlyAvailable <Boolean> ] [-JobGroup <Guid> ] [-JobVariable <String> ] [-LinuxAdministratorSSHKey <SSHKey> ] [-LinuxDomainName <String> ] [-MemoryMB <Int32> ] [-MemoryPerVirtualNumaNodeMaximumMB <Int32> ] [-MemoryWeight <Int32> ] [-MonitorMaximumCount <Int32> ] [-MonitorMaximumResolution <String> ] [-NumaIsolationRequired <Boolean> ] [-OperatingSystem <OperatingSystem> ] [-Owner <String> ] [-PROTipID <Guid> ] [-ReturnImmediately] [-RunAsynchronously] [-SkipInstallVirtualizationGuestServices] [-StartAction {NeverAutoTurnOnVM | AlwaysAutoTurnOnVM | TurnOnVMIfRunningWhenVSStopped} ] [-StartVM] [-StopAction {SaveVM | TurnOffVM | ShutdownGuestOS} ] [-UseLocalVirtualHardDisk] [-UserRole <UserRole> ] [-VirtualNumaNodesPerSocketMaximum <Byte> ] [-VirtualVideoAdapterEnabled <Boolean> ] [ <CommonParameters>]

Parameter Set: NewVmFromVm

New-SCVirtualMachine [-Name] <String> -Path <String> -VM <VM> -VMHost <Host> [-CPUCount <Byte> ] [-CPUCyclesLimitPercent <UInt16> ] [-CPUCyclesReservePercent <UInt16> ] [-

---

CPULimitForMigration <Boolean> ] [-CPULimitFunctionality <Boolean> ] [-  
CPUPerVirtualNumaNodeMaximum <Byte> ] [-CPURelativeWeight <Int32> ] [-CPUType  
<ProcessorType> ] [-DelayStartSeconds <Int32> ] [-Description <String> ] [-  
DynamicMemoryBufferPercentage <Int32> ] [-DynamicMemoryEnabled <Boolean> ] [-  
DynamicMemoryMaximumMB <Int32> ] [-DynamicMemoryMinimumMB <UInt16> ] [-HardwareProfile  
<HardwareProfile> ] [-JobGroup <Guid> ] [-JobVariable <String> ] [-LinuxAdministratorSSHKey  
<SSHKey> ] [-LinuxDomainName <String> ] [-MemoryMB <Int32> ] [-  
MemoryPerVirtualNumaNodeMaximumMB <Int32> ] [-MemoryWeight <Int32> ] [-MonitorMaximumCount  
<Int32> ] [-MonitorMaximumResolution <String> ] [-NumaIsolationRequired <Boolean> ] [-  
OperatingSystem <OperatingSystem> ] [-Owner <String> ] [-PROTipID <Guid> ] [-  
ReturnImmediately] [-RunAsynchronously] [-SkipInstallVirtualizationGuestServices] [-  
StartAction {NeverAutoTurnOnVM | AlwaysAutoTurnOnVM | TurnOnVMIIfRunningWhenVSSStopped} ] [-  
StartVM] [-StopAction {SaveVM | TurnOffVM | ShutdownGuestOS} ] [-UseLocalVirtualHardDisk] [-  
UserRole <UserRole> ] [-VirtualNumaNodesPerSocketMaximum <Byte> ] [-  
VirtualVideoAdapterEnabled <Boolean> ] [ <CommonParameters>]

Parameter Set: NewVmFromVmCloud

New-SCVirtualMachine [-Name] <String> -Cloud <Cloud> -VM <VM> [-CapabilityProfile  
<CapabilityProfile> ] [-CPUCount <Byte> ] [-CPUCyclesLimitPercent <UInt16> ] [-  
CPUCyclesReservePercent <UInt16> ] [-CPULimitForMigration <Boolean> ] [-  
CPULimitFunctionality <Boolean> ] [-CPUPerVirtualNumaNodeMaximum <Byte> ] [-  
CPURelativeWeight <Int32> ] [-CPUType <ProcessorType> ] [-DelayStartSeconds <Int32> ] [-  
Description <String> ] [-DynamicMemoryBufferPercentage <Int32> ] [-DynamicMemoryEnabled  
<Boolean> ] [-DynamicMemoryMaximumMB <Int32> ] [-DynamicMemoryMinimumMB <UInt16> ] [-  
HardwareProfile <HardwareProfile> ] [-JobGroup <Guid> ] [-JobVariable <String> ] [-  
LinuxAdministratorSSHKey <SSHKey> ] [-LinuxDomainName <String> ] [-MemoryMB <Int32> ] [-  
MemoryPerVirtualNumaNodeMaximumMB <Int32> ] [-MemoryWeight <Int32> ] [-MonitorMaximumCount  
<Int32> ] [-MonitorMaximumResolution <String> ] [-NumaIsolationRequired <Boolean> ] [-  
OperatingSystem <OperatingSystem> ] [-Owner <String> ] [-PROTipID <Guid> ] [-  
ReturnImmediately] [-RunAsynchronously] [-SelfServiceRole <SelfServiceUserRole> ] [-  
SkipInstallVirtualizationGuestServices] [-StartAction {NeverAutoTurnOnVM |  
AlwaysAutoTurnOnVM | TurnOnVMIIfRunningWhenVSSStopped} ] [-StartVM] [-StopAction {SaveVM |  
TurnOffVM | ShutdownGuestOS} ] [-StoreToLibrary] [-UseLocalVirtualHardDisk] [-UserRole  
<UserRole> ] [-VirtualNumaNodesPerSocketMaximum <Byte> ] [-VirtualVideoAdapterEnabled  
<Boolean> ] [ <CommonParameters>]

Parameter Set: NewVmFromVmConfig

New-SCVirtualMachine [-Name] <String> -VMConfiguration <BaseVMConfiguration> [-AnswerFile  
<Script> ] [-BlockDynamicOptimization <Boolean> ] [-CapabilityProfile <CapabilityProfile> ]  
[-Cloud <Cloud> ] [-ComputerName <String> ] [-CPUCount <Byte> ] [-CPUCyclesLimitPercent  
<UInt16> ] [-CPUCyclesReservePercent <UInt16> ] [-CPULimitForMigration <Boolean> ] [-  
CPULimitFunctionality <Boolean> ] [-CPUPerVirtualNumaNodeMaximum <Byte> ] [-  
CPURelativeWeight <Int32> ] [-CPUType <ProcessorType> ] [-DelayStartSeconds <Int32> ] [-  
Description <String> ] [-Domain <String> ] [-DomainJoinCredential <VMCredentials> ] [-  
DynamicMemoryBufferPercentage <Int32> ] [-DynamicMemoryEnabled <Boolean> ] [-  
DynamicMemoryMaximumMB <Int32> ] [-DynamicMemoryMinimumMB <UInt16> ] [-FullName <String> ]  
[-GuestOSProfile <GuestOSProfile> ] [-GuiRunOnceCommands <String[]> ] [-HardwareProfile  
<HardwareProfile> ] [-JobGroup <Guid> ] [-JobVariable <String> ] [-LinuxAdministratorSSHKey  
<SSHKey> ] [-LinuxDomainName <String> ] [-LocalAdministratorCredential <VMCredentials> ] [-  
MemoryMB <Int32> ] [-MemoryPerVirtualNumaNodeMaximumMB <Int32> ] [-MemoryWeight <Int32> ] [-  
MergeAnswerFile <Boolean> ] [-MonitorMaximumCount <Int32> ] [-MonitorMaximumResolution  
<String> ] [-NumaIsolationRequired <Boolean> ] [-OperatingSystem <OperatingSystem> ] [-



---

```
OrganizationName <String> ] [-Owner <String> ] [-ProductKey <String> ] [-PROTipID <Guid> ]
[-ReturnImmediately] [-RunAsynchronously] [-SelfServiceRole <SelfServiceUserRole> ] [-
SkipInstallVirtualizationGuestServices] [-StartAction {NeverAutoTurnOnVM |
AlwaysAutoTurnOnVM | TurnOnVMIfRunningWhenVSSStopped} ] [-StartVM] [-StopAction {SaveVM |
TurnOffVM | ShutdownGuestOS} ] [-StoreToLibrary] [-TimeZone <Int32> ] [-
UseLocalVirtualHardDisk] [-UserRole <UserRole> ] [-VirtualNumaNodesPerSocketMaximum <Byte> ]
[-VirtualVideoAdapterEnabled <Boolean> ] [-Workgroup <String> ] [ <CommonParameters>]
```

Parameter Set: NewVmFromVmConfigScaleOut

```
New-SCVirtualMachine [-Name] <String> -VMConfigurationScaleOut <BaseVMConfiguration> [-
CPUCount <Byte> ] [-CPUCyclesLimitPercent <UInt16> ] [-CPUCyclesReservePercent <UInt16> ] [-
CPULimitForMigration <Boolean> ] [-CPULimitFunctionality <Boolean> ] [-
CPUPerVirtualNumaNodeMaximum <Byte> ] [-CPURelativeWeight <Int32> ] [-CPUType
<ProcessorType> ] [-DelayStartSeconds <Int32> ] [-Description <String> ] [-
DynamicMemoryBufferPercentage <Int32> ] [-DynamicMemoryEnabled <Boolean> ] [-
DynamicMemoryMaximumMB <Int32> ] [-DynamicMemoryMinimumMB <UInt16> ] [-HardwareProfile
<HardwareProfile> ] [-JobGroup <Guid> ] [-JobVariable <String> ] [-LinuxAdministratorSSHKey
<SSHKey> ] [-LinuxDomainName <String> ] [-MemoryMB <Int32> ] [-
MemoryPerVirtualNumaNodeMaximumMB <Int32> ] [-MemoryWeight <Int32> ] [-MonitorMaximumCount
<Int32> ] [-MonitorMaximumResolution <String> ] [-NumaIsolationRequired <Boolean> ] [-
OperatingSystem <OperatingSystem> ] [-Owner <String> ] [-PROTipID <Guid> ] [-
ReturnImmediately] [-RunAsynchronously] [-SkipInstallVirtualizationGuestServices] [-
StartAction {NeverAutoTurnOnVM | AlwaysAutoTurnOnVM | TurnOnVMIfRunningWhenVSSStopped} ] [-
StopAction {SaveVM | TurnOffVM | ShutdownGuestOS} ] [-UseLocalVirtualHardDisk] [-UserRole
<UserRole> ] [-VirtualNumaNodesPerSocketMaximum <Byte> ] [-VirtualVideoAdapterEnabled
<Boolean> ] [ <CommonParameters>]
```

## Detailed Description

The New-SCVirtualMachine cmdlet creates a virtual machine to be managed by Virtual Machine Manager (VMM). You can create a virtual machine from an existing stopped virtual machine deployed on a host, from an existing virtual machine stored in the VMM library, from a virtual machine template, from an existing virtual hard disk that already contains an operating system, or from a blank virtual hard disk. For example, you can create a new virtual machine from an existing hard disk that contains a third-party operating system, such as Linux.

New in System Center 2012 Service Pack 1 (SP1), you can create a virtual machine by using a differencing disk. For more information about differencing disks, see New-SCVirtualDiskDrive.

When you deploy a new virtual machine to a Hyper-V host, you can specify a location for the virtual machine files, or use the default path of <C>:\ProgramData\Microsoft\Windows\Hyper-V

When you deploy a new virtual machine on a VMware ESX host or Citrix XenServer host, there is no default path to store the virtual machine files, so you must specify the path when you create the virtual machine.

As an alternative to using New-SCVirtualMachine, you can also use the following cmdlets to create a new virtual machine:

- New-SCP2V - creates a new virtual machine from an existing physical

---

machine (a P2V conversion). For more information, type: "Get-Help New-SCP2V -detailed".

- New-SCV2V - creates a new virtual machine from an existing virtual machine, such as a virtual machine created in VMWare (a V2V conversion). For more information, type: "Get-Help New-SCV2V -detailed".

## Parameters

### **-AnswerFile<Script>**

Specifies a script object stored in the VMM library to use as an answer file. The name of the answer file script depends on the operating system that you want to install on a virtual machine:

ANSWER FILE    GUEST OS TO INSTALL ON VM

-----    -----

Sysprep.inf    Windows XP, Windows Server 2000, or Windows Server 2003

Unattend.xml    Windows Vista, Windows 7, or Windows Server 2008

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-BlockDynamicOptimization<Boolean>**

Indicates whether dynamic optimization is blocked for a virtual machine.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-CapabilityProfile<CapabilityProfile>**

Specifies a capability profile object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-Cloud<Cloud>**

Specifies a private cloud object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-ComputerName<String>**

Specifies the name of a computer that VMM can uniquely identify on your network. Valid formats are: FQDN, IPv4 or IPv6 address, or NetBIOS name.

NOTE: See the examples for a specific cmdlet to determine how that cmdlet specifies the computer name.

|                        |       |
|------------------------|-------|
| Aliases                | none  |
| Required?              | false |
| Position?              | named |
| Default Value          | none  |
| Accept Pipeline Input? | false |

|                             |       |
|-----------------------------|-------|
| Accept Wildcard Characters? | false |
|-----------------------------|-------|

### **-ComputerTier<ComputerTier>**

Specifies a computer tier object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-CPUCount<Byte>**

Specifies the number of CPUs on a virtual machine, on a hardware profile, or on a template. See the examples for a specific cmdlet to determine how that cmdlet uses this parameter.

TYPE OF HOST    NUMBER OF PROCESSORS

-----    -----

Hyper-V        Up to 4 CPUs per VM; varies by guest OS

VMware ESX    Up to 4 CPUs per VM for any supported guest OS

Exception: 1 CPU on a VM running Windows NT 4.0

Citrix XenServer Up to 8 CPUs per VM; varies by guest OS

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-CPUCyclesLimitPercent<UInt16>**

Specifies the limit of CPU cycles in percent.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-CPUCyclesReservePercent<UInt16>**

Specifies the reserve CPU cycles in percent.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-CPULimitForMigration<Boolean>**

Limits, when set to \$True, processor features for the specified virtual machine in order to enable migration to a physical computer that has a different version of the same processor as the source computer. VMM does not support migrating virtual machines between physical computers that have processors from different manufacturers.

|                             |                      |
|-----------------------------|----------------------|
| Aliases                     | LimitCPUForMigration |
| Required?                   | false                |
| Position?                   | named                |
| Default Value               | none                 |
| Accept Pipeline Input?      | false                |
| Accept Wildcard Characters? | false                |

---

## **-CPULimitFunctionality<Boolean>**

Enables running an older operating system (such as Windows NT 4.0) on a virtual machine deployed on a Hyper-V host or on a VMware ESX host by providing limited CPU functionality for the virtual machine.

|                             |                       |
|-----------------------------|-----------------------|
| Aliases                     | LimitCPUFunctionality |
| Required?                   | false                 |
| Position?                   | named                 |
| Default Value               | none                  |
| Accept Pipeline Input?      | false                 |
| Accept Wildcard Characters? | false                 |

## **-CPUPerVirtualNumaNodeMaximum<Byte>**

Specifies the maximum number of CPUs allowed for each virtual NUMA node.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-CPURelativeWeight<Int32>**

Specifies the amount of CPU resources on a host that this virtual machine can use relative to other virtual machines on the same host. A virtual machine with a higher setting is allocated more CPU resources than a virtual machine with a lower setting.

TYPE OF HOST    RANGE OF RELATIVE VALUES

-----    -----

Hyper-V        1 to 10000

VMware ESX    2000 = High

1500 = Above Normal

1000 = Normal (default)

750 = Below Normal

---

500 = Low

1 to 1000000 = Custom

The VMware term for these values is "shares."

Citrix XenServer 1 to 65536, normal is 256

NOTE: See the examples for a specific cmdlet to determine how that cmdlet uses this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-CPUType<ProcessorType>**

Specifies the type of CPU for a virtual machine. To retrieve a list of all CPU types that are available for use in virtual machines in a VMM environment, type: "Get-SCCPUType"

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-DelayStartSeconds<Int32>**

Specifies the number of seconds to wait after the virtualization service starts before automatically starting a virtual machine. This delay is used to stagger the startup time of multiple virtual machines to help reduce the demand on the physical computer's resources. A typical setting might be 30 to 60 seconds.

TYPE OF HOST      MAXIMUM CONFIGURABLE DELAY

-----      -----

Hyper-V      1000000000 seconds (277777 hours)

VMware ESX      65535 seconds (18 hours)

Citrix XenServer      Does not apply to XenServer virtual machines

---

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Description<String>**

States a description for the specified object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Domain<String>**

Specifies a fully qualified domain name (FQDN) for an Active Directory domain.

Example format: -Domain "Domain01.Corp.Contoso.com"

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |



---

## **-DomainJoinCredential<VMMCredential>**

Specifies the user name and password of an account with permission to join a computer to the domain. A limited rights account should be used for joining computers (either physical or virtual) to the domain.

Example format for a PS credential:

```
$DomainJoinCredential = Get-Credential
```

```
-Domain "ThisDomain.Corp.Contoso.com" -DomainJoinCredential $DomainJoinCredential
```

Example format for a Run As account:

```
$DomainJoinCredential = Get-SCRunAsAccount -Name "RunAsAcct01"
```

```
-Domain "ThisDomain.Corp.Contoso.com" -DomainJoinCredential $DomainJoinCredential
```

NOTE: You can use the DomainJoinCredential parameter to specify credentials (on a VMHostProfile) for joining a physical host computer to the domain, or to specify credentials (on a new or existing template, on a new or existing guest operating system profile, or on a new virtual machine) for joining a virtual machine to the domain.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-DRProtectionRequired<Boolean>**

This parameter is reserved for future use.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-DynamicMemoryBufferPercentage<Int32>**

Specifies the percentage of memory above a virtual machine's current memory allocation which the host should try to reserve as a buffer. The default value is 20.

---

Example format: -DynamicMemoryTargetBufferPercentage 20

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-DynamicMemoryEnabled<Boolean>**

Enables, when set to \$True, dynamic memory for virtual machines. You can enable dynamic memory directly on a virtual machine, or on a template or hardware profile that will be used to create virtual machines. The default value is False.

REQUIRED: You can enable dynamic memory for a virtual machine only if that virtual machine is deployed on a host running Windows Server 2008 SP1 or later or if the virtual machine is stored in a library in a stopped state (hardware changes to a stored virtual machine can only be made if the virtual machine does not have snapshots). Enabling dynamic memory on a virtual machine stored in a library will limit placement of that machine to hosts running Windows Server 2008 SP1 or later.

Example format: -DynamicMemoryEnabled \$True

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-DynamicMemoryMaximumMB<Int32>**

Specifies the maximum amount of memory that can be allocated to a virtual machine if dynamic memory is enabled. The default value is 65536.

REQUIRED: You can enable dynamic memory for a virtual machine only if that virtual machine is deployed on a host running Windows Server 2008 R2 SP1 or later or if the virtual machine is stored in a library in a stopped state (hardware changes to a stored virtual machine can only be made if the virtual machine does not have snapshots). Enabling dynamic memory on a virtual machine stored in a library will limit placement of that machine to hosts running Windows Server 2008 R2 SP1 or later.

---

Example format: -DynamicMemoryMaximumMB 1024

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-DynamicMemoryMinimumMB<UInt16>**

Specifies the minimum amount of memory that can be allocated to a virtual machine if dynamic memory is enabled. The default value is 65536.

REQUIRED: You can enable dynamic memory for a virtual machine only if that virtual machine is deployed on a host running Windows Server 2008 R2 SP1 or later or if the virtual machine is stored in a library in a stopped state (hardware changes to a stored virtual machine can only be made if the virtual machine does not have snapshots). Enabling dynamic memory on a virtual machine stored in a library will limit placement of that machine to hosts running Windows Server 2008 R2 SP1 or later.

Example format: -DynamicMemoryMinimumMB 1024

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-FullName<String>**

Specifies the name of the person in whose name a virtual machine is registered.

|               |       |
|---------------|-------|
| Aliases       | none  |
| Required?     | false |
| Position?     | named |
| Default Value | none  |

|                             |       |
|-----------------------------|-------|
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-GuestOSProfile<GuestOSProfile>**

Specifies a guest operating system profile object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-GuiRunOnceCommands<String[]>**

Specifies one or more commands to add to the [GuiRunOnce] section of an unattended answer file (such as Unattend.xml). Use single quotes around each string enclosed in double quotes.

Example format:

```
-GuiRunOnceCommands "C:\APF\APFPostSysPrepCopy.cmd PARAMS1",
"C:\APF\APFPostSysPrepCopy.cmd PARAMS1"
```

For information about how Windows PowerShell uses quotes, type: "Get-Help about\_Quoting\_Rules"

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-HardwareProfile<HardwareProfile>**

Specifies a hardware profile object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-HighlyAvailable<Boolean>**

Specifies that a virtual machine will be placed on a Hyper-V host that is part of a host cluster. Configure this setting on a virtual machine, or on a template or hardware profile that will be used to create virtual machines.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobGroup<Guid>**

Specifies an identifier for a series of commands that will run as a set just before the final command that includes the same job group identifier runs.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-LibraryServer<LibraryServer>**

Specifies a VMM library server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-LinuxAdministratorSSHKey<SSHKey>**

Specifies the public key file for a Linux Ssh Key.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-LinuxDomainName<String>**

Specifies a fully qualified domain name (FQDN) to be used in conjunction with Linux operating system specialization.

Example format: -LinuxDomainName "Domain01.Corp.Contoso.com"

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-LocalAdministratorCredential<VMNCredential>**

Specifies the user name and password for the Local Administrator account (or Linux root account in the case of a Linux compatible Guest Operating System profile).

Specifying credentials on a new or existing template, on a new or existing guest operating system profile, or on a new virtual machine overrides any existing Administrator password.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-MemoryMB<Int32>**

Specifies, in megabytes (MB), the amount of random access memory (RAM) on the host that is allocated to a virtual machine. The default value is 512 MB. For a virtual machine on which dynamic memory is enabled (on a host running Windows Server 2008 R2 SP1 or later), use MemoryMB to specify the startup memory value.

TYPE OF HOST              MAXIMUM HOST MEMORY ASSIGNABLE TO VM

-----                      -----

Hyper-V                      Up to 65536 MB RAM per virtual machine

VMware ESX Server 3.0.x Up to 16384 MB RAM per virtual machine

---

VMware ESX Server 3.5.x Up to 65532 MB RAM per virtual machine

Citrix XenServer Up to 32265 MB RAM per VM

Example format: -MemoryMB 1024

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-MemoryPerVirtualNumaNodeMaximumMB<Int32>**

Specifies the maximum amount of memory, in MB, that each virtual NUMA node is allowed.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-MemoryWeight<Int32>**

Indicates the priority in allocating memory to a virtual machine, relative to other virtual machines on the same host. A virtual machine with a higher setting is allocated more memory resources than a virtual machine with a lower setting.

For a host running Windows Server 2008 R2 SP1 or later, 5000 = Normal, 10000 = High, 0 = Low, 1 to 10000 = Custom.

NOTE: See the examples for a specific cmdlet to determine how that cmdlet uses this parameter.

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | false |
| Position? | named |



|                             |       |
|-----------------------------|-------|
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-MergeAnswerFile<Boolean>**

Specifies that the cmdlet merge the specified answer file with the specified guest operating system settings. The default value is TRUE. This parameter is used by the VMM console. You do not need to use this parameter at the command prompt.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-MonitorMaximumCount<Int32>**

Specifies the maximum number of monitors supported by a virtual video adapter.

Example format: -MonitorMaximumCount 3

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-MonitorMaximumResolution<String>**

Specifies, as a string, the value that represents the maximum possible monitor resolution of a virtual video adapter. Valid values are: "1024x768", "1280x1024", "1600x1200", "1920x1200". Default value: "1280x1024"

Example format: -MonitorResolutionMaximum "1600x1200"

---

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Name<String>**

Specifies the name of a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | 1     |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-NumaIsolationRequired<Boolean>**

Indicates whether NUMA isolation is required.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-OperatingSystem<OperatingSystem>**

Specifies the type of operating system for a virtual machine. To list the names of all available operating systems in VMM, type: "Get-SCOperatingSystem".

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-OrganizationName<String>**

Specifies the name of the organization for the person in whose name a virtual machine is registered.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-Owner<String>**

Specifies the owner of a VMM object in the form of a valid domain user account.

Example format: -Owner "Contoso\ReneeLo"

Example format: -Owner "ReneeLo@Contoso"

|                        |       |
|------------------------|-------|
| Aliases                | none  |
| Required?              | false |
| Position?              | named |
| Default Value          | none  |
| Accept Pipeline Input? | false |

|                             |       |
|-----------------------------|-------|
| Accept Wildcard Characters? | false |
|-----------------------------|-------|

## **-Path<String>**

Specifies the destination path for the operation.

Example formats:

Local path -Path "F:\"

UNC path -Path "\\Library\Templates"

Volume GUID path -Path "\\?\Volume{4703c1ea-8ae7-11db-b473-00123f7603e3}\"

VMware ESX path -Path "[storage1]\MyVMwareFolderForVMs\MyVM.vmx"

Citrix XenServer path - Path "Local storage[99b6212f-b63d-c676-25f9-d6c460992de7]"

Wildcards are supported for "Get" cmdlets and when you specify the UNC path:

Example format:

UNC path -Path "\\VMHostServer\MyVMs\\*VM\*"

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-ProductKey<String>**

Specifies a product key. The product key is a 25-digit number that identifies the product license. A product key can be used to register VMM or an operating system to be installed on a virtual machine or host.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-ReturnImmediately**

Indicates that control is returned to the calling process immediately, before the job is created.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-SelfServiceRole<SelfServiceUserRole>**

Specifies the self-service role with permission to access the virtual machine.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-SharePath<String>**

Specifies a path to a valid library share on an existing library server that uses a Universal Naming Convention (UNC) path.

Example format: `-SharePath "\\LibServer01\LibShare"`

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-SkipInstallVirtualizationGuestServices**

Skips the installation of virtualization guest services on a virtual machine. By default, this parameter is set to `$False` and VMM installs the appropriate virtualization guest service automatically. For a virtual machine on a Hyper-V host, the virtualization guest service is called Integration Components (VMGuest.iso). For a virtual machine on a XenServer host, the virtualization guest service is called Citrix Tools for Virtual Machines (xs-tools.iso). Virtual machines on a VMware ESX host do not use a virtualization guest service.

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | false |
| Position? | named |

|                             |       |
|-----------------------------|-------|
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-StartAction<VMStartAction>**

Specifies the behavior of a virtual machine when the virtualization service (Hyper-V, VMware, or XenServer) starts. Valid values are: AlwaysAutoTurnOnVM, NeverAutoTurnOnVM, TurnOnVMIfRunningWhenVSStopped.

The acceptable values for this parameter are:

|                                |  |
|--------------------------------|--|
| NeverAutoTurnOnVM              |  |
| AlwaysAutoTurnOnVM             |  |
| TurnOnVMIfRunningWhenVSStopped |  |

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-StartVM**

Specifies that the virtual machine starts when it arrives at the destination host.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-StopAction<VMStopAction>**

Specifies the behavior of the virtual machine when the virtualization service (Hyper-V, VMware, or XenServer) stops. Valid values are: SaveVM, TurnOffVM, ShutdownGuestOS.

The acceptable values for this parameter are:

|                 |  |
|-----------------|--|
| SaveVM          |  |
| TurnOffVM       |  |
| ShutdownGuestOS |  |

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-StoreToLibrary**

Indicates that the virtual machine should be stored in the VMM library.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-TimeZone<Int32>**

Specifies a number (an index) that identifies a geographical region that shares the same standard time. For a list of time zone indexes, see "Microsoft Time Zone Index Values" at:



---

<http://go.microsoft.com/fwlink/?LinkId=120935>. If no time zone is specified, the default time zone used for a virtual machine is the same time zone setting that is on the virtual machine host.

Example format to specify the GMT Standard Time zone: `-TimeZone 085`

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-UseLocalVirtualHardDisk**

Verifies that the VHD file (or files) to be used to create the virtual machine exist and are stored on the destination host.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-UserRole<UserRole>**

Specifies a user role object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-VirtualHardDisk<StandaloneVirtualHardDisk>**

Specifies a virtual hard disk object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-VirtualNumaNodesPerSocketMaximum<Byte>**

Specifies the maximum number of NUMA nodes allowed for each socket.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VirtualVideoAdapterEnabled<Boolean>**

Enables, when set to \$True, the Microsoft Synthetic 3D Virtual Video Adapter for virtual machines. You can enable the Virtual Video Adapter directly on a virtual machine, or on a template or hardware profile that will be used to create virtual machines.

REQUIRED: You can enable the Microsoft Synthetic 3D Virtual Video Adapter for a virtual machine only if that virtual machine is deployed on a host running Windows Server 2008 R2 SP1 (with the Remote Desktop Services role and Remote Desktop Virtual Graphics role service installed) or later or if the virtual machine is stored in a library in a stopped state (hardware changes to a stored virtual machine can only be made if the virtual machine does not have snapshots). Enabling the Microsoft Synthetic 3D Virtual Video Adapter on a virtual machine stored in a library will limit placement of that machine to hosts running Windows Server 2008 R2 SP1 (with the Remote Desktop Services role and Remote Desktop Virtual Graphics role service installed) or later.

Example format: -VirtualVideoAdapterEnabled \$TRUE

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VM<VM>**

Specifies a virtual machine object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-VMConfiguration<BaseVMConfiguration>**

Specifies a virtual machine configuration object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMConfigurationScaleOut<BaseVMConfiguration>**

Specifies a virtual machine configuration object used when scaling out a service.

|                             |                         |
|-----------------------------|-------------------------|
| Aliases                     | ScaleOutVMConfiguration |
| Required?                   | true                    |
| Position?                   | named                   |
| Default Value               | none                    |
| Accept Pipeline Input?      | false                   |
| Accept Wildcard Characters? | false                   |

### **-VMHost<Host>**

Specifies a virtual machine host object. VMM supports Hyper-V hosts, VMware ESX hosts, and Citrix XenServer hosts.

For more information about each type of host, type: "Get-Help Add-SCVMHost -detailed". See the examples for a specific cmdlet to determine how that cmdlet uses this parameter.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-VMTemplate<Template>**

Specifies a VMM template object used to create virtual machines.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | Template       |
| Required?                   | true           |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-Workgroup<String>**

Specifies on a new or existing template, on a new or existing guest operating system profile, or on a new virtual machine the name of the workgroup to which you want to join a virtual machine. You can use this parameter to override the existing value on a template or on a guest operating system profile.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **VirtualMachine**

---

## Notes

- Requires a VMM virtual hard disk object, virtual machine template object, or virtual machine object, which can be retrieved by using the Get-SCVirtualHardDisk, Get-SCVMTemplate, or Get-SCVirtualMachine cmdlets, respectively.

## Examples

### 1: Create a virtual machine from a virtual hard disk and deploy it on a host.

The first command gets the virtual hard disk object named "Blank Disk - Large" from the VMM library and stores the object in the \$VHD variable.

The second command gets the host object named VMHost01 and stores the object in the \$VMHost object.

The last command creates a virtual machine named VM01 from the virtual hard disk stored in \$VHD (in this case, "Blank Disk - Large") and deploys the new virtual machine in C:\VirtualMachinePath on VMHost01. The RunAsynchronously parameter returns control to the shell immediately, before the command completes.

```
PS C:\> $VHD = Get-SCVirtualHardDisk -Name "Blank Disk - Large"
PS C:\> $VMHost = Get-SCVMHost -ComputerName "VMHost01.Contoso.com"
PS C:\> New-SCVirtualMachine -Name "VM01" -VirtualHardDisk $VHD -VMHost $VMHost -Path
"C:\VirtualMachinePath" -RunAsynchronously
```

### 2: Create a virtual machine from a virtual machine template and deploy it on a host.

The first command gets the virtual machine template object named "WindowsServer2008R2" and stores the object in the \$Template variable.

The second command gets the host object named VMHost02 and stores the object in the \$VMHost variable.

The last command creates a new virtual machine from the virtual machine template stored in \$Template, names the virtual machine VM02, and deploys the virtual machine on host VMHost02, storing its files at C:\VirtualMachinePath. When the virtual machine is created, the following properties are customized: the computer name for the virtual machine (Server01), the name of the person to whom the virtual machine is registered (Renee Lo), the organization name (Contoso), and the product key (substitute your product key for the Xs). Using the RunAsynchronously parameter returns control to the shell immediately, before the job completes.

```
PS C:\> $VMTemplate = Get-SCVMTemplate -VMMServer "VMMServer01.Contoso.com" | where {$_.Name -eq "WindowsServer2008R2"}
PS C:\> $VMHost = Get-SCVMHost -ComputerName "VMHost02.Contoso.com"
```

---

```
PS C:\> New-SCVirtualMachine -VMTemplate $VMTemplate -Name "VM02" -VMHost $VMHost -Path
"C:\VirtualMachinePath" -RunAsynchronously -ComputerName "Server01" -FullName "Renee Lo" -
OrgName "Contoso" -ProductKey "XXXXX-XXXXX-XXXXX-XXXXX-XXXXX"
```

### 3: Create a virtual machine by cloning an existing virtual machine.

The first command gets the virtual machine object named VM03 and stores the object in the \$VM variable.

The second command gets the host object named VMHost03 and stores the object in the \$VMHost variable.

The last command checks whether virtual machine VM01 is in a powered off state. If the virtual machine is powered off, the command creates a virtual machine named VM03 from VM01 and deploys the new virtual machine on VMHost03 in C:\VirtualMachinePath. The RunAsynchronously parameter returns control to the shell immediately, before the command completes.

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM01"
PS C:\> $VMHost = Get-SCVMHost -ComputerName "VMHost01.Contoso.com"
PS C:\> if($VM.Status -eq "PowerOff"){New-SCVirtualMachine -Name "VM03" -VM $VM -VMHost
$VMHost -Path "C:\VirtualMachinePath" -RunAsynchronously}
```

### 4. Create a virtual machine from a virtual machine stored in the library.

The first command gets the host object named VMHost04 and stores the object in the \$VMHost variable.

The second command gets the virtual machine object named StoredVM01, which is located at the specified path on library server LibServer01, and stores the virtual machine object in the \$VM variable.

The last command creates a new virtual machine named VM04 from StoredVM01, provides the new virtual machine with 1024 MB of memory, and deploys it at the specified path. The command also provides a description and owner, and specifies that the start action for the virtual machine is set to never turn on automatically and that the stop action is to save the virtual machine. Using the RunAsynchronously parameter returns control to the shell immediately, before the command completes.

```
PS C:\> $VMHost = Get-SCVMHost -ComputerName "VMHost04"
PS C:\> $VM = Get-SCVirtualMachine -Name "StoredVM01" | where {$_.LibraryServer.Name -eq
"LibServer01.Contoso.com"} | where {$_.Location -eq
"\\LibServer01.Consoso.com\MSSCVMLibrary\StoredVM01"}
PS C:\> New-SCVirtualMachine -VM $VM -Name "VM04" -Description "New VM from VM stored in
Library" -Owner "Contoso\ReneeLo" -VMHost $VMHost -Path "C:\VirtualMachinePath" -
RunAsynchronously -StartAction NeverAutoTurnOnVM -StopAction SaveVM -MemoryMB 1024
```

---

## 5. Create a highly available virtual machine.

The first command creates a new GUID string and stores it in `$VMGuid`. This GUID is a job group ID that functions as an identifier that groups subsequent commands that include this identifier into a single job group.

The second command stores the string "HAVM01" in `$VMName`, and is used to name the new virtual machine.

The third command creates a virtual network adapter with a dynamic MAC address and with VLAN disabled. Because the command uses the `JobGroup` variable, the network adapter is not created until just before the `New-SCVirtualMachine` cmdlet in the last command runs.

The fourth command creates an IDE virtual DVD drive connected to the second channel and the first slot. Because the command uses the `JobGroup` variable, the virtual DVD drive is not created until just before the `New-SCVirtualMachine` cmdlet in the last command runs.

The fifth command creates a hardware profile and specifies values for the profile name, owner, CPU count, memory, and bootorder. The command disables `NumLock` as well as limited CPU functionality, which is not needed because this virtual machine will not have an older operating system, and designates that the virtual machine created by using this hardware profile will be highly available. Because the command uses the `JobGroup` variable, the hardware profile is not created until just before the `New-SCVirtualMachine` cmdlet in the last command runs.

The sixth command creates an IDE virtual disk drive with a storage capacity of 4 GB on the first channel and first slot. Because the command uses the `JobGroup` variable, the new virtual disk drive is not created until just before the `New-SCVirtualMachine` cmdlet in the last command runs.

The seventh command gets a virtual machine host object by name (`VMMHANode02`) and stores the object in `$VMHost`. This host is one node of a host cluster that is managed by VMM.

The eighth command gets the hardware profile object named `HWProfile02`, which was created in the fifth command, and stores the object in the `$HardwareProfile` variable.

The ninth command gets an operating system object by name and stores the object in `$OperatingSystem`.

The last command uses the `New-SCVirtualMachine` cmdlet and the `JobGroup` parameter to create a new highly available virtual machine named `HAVM01` by using the objects created and obtained in the preceding commands. The command also uses the `Path` parameter to specify the location where the virtual machine is stored, which must be a cluster-migratable LUN. Additionally, the command specifies that the virtual machine is not started automatically when the host starts and that the virtual machine is put into a saved state when the virtualization service stops. Using the `RunAsynchronously` parameter returns control to the command shell immediately, before the command completes.

```
PS C:\> $JobGuid = [System.Guid]::NewGuid().ToString()
```

```
PS C:\> $VMName = "HAVM01"
```

```
PS C:\> New-SCVirtualNetworkAdapter -JobGroup $JobGuid -PhysicalAddressType Dynamic -
VLANEnabled $False
```

```
PS C:\> New-SCVirtualDVDDrive -JobGroup $JobGuid -Bus 1 -LUN 0
```

```
PS C:\> New-SCHardwareProfile -Owner "Contoso\ReneeLo" -Name "HWProfile02" -CPUCount 1 -
MemoryMB 512 -HighlyAvailable $True -NumLock $False -BootOrder "CD", "IdeHardDrive",
"PxeBoot", "Floppy" -LimitCPUFunctionality $False -JobGroup $JobGuid
```

```
PS C:\> New-SCVirtualDiskDrive -IDE -Bus 0 -LUN 0 -JobGroup $JobGuid -Size 40960 -Dynamic -
Filename "HAVM01_disk_1.vhd"
```



---

```

PS C:\> $VMHost = Get-SCVMHost | where {$_.Name -eq "VMMHNode02.Contoso.com"}
PS C:\> $HardwareProfile = Get-SCHardwareProfile | where {$_.Name -eq "HWProfile02"}
PS C:\> $OperatingSystem = Get-SCOperatingSystem | where {$_.Name -eq "64-bit edition of
Windows Server 2008 R2 Datacenter"}
PS C:\> New-SCVirtualMachine -Name $VMName -Description "" -VMMServer
"VMMServer01.Contoso.com" -Owner "Contoso\ReneeLo" -VMHost $VMHost -Path "R:\\" -
HardwareProfile $HardwareProfile -JobGroup $JobGuid -OperatingSystem $OperatingSystem -
RunAsynchronously -StartAction NeverAutoTurnOnVM -StopAction SaveVM

```

## 6: Use an existing VHD file on the destination host to create a new virtual machine from a template.

The first command generates a globally unique identifier (GUID) and stores the GUID string in the \$JobGroupID variable. The job group ID functions as an identifier that groups subsequent commands that include this identifier into a single job group.

The second command gets the virtual machine template object named VMTemplate01 and stores the object in the \$VMTemplate variable. VMTemplate01 is assumed to have a virtual disk drive on IDE Bus 0 LUN 0 that contains a virtual hard disk.

The third command gets the host object named VMHost06 and stores the object in the \$VMHost variable.

The fourth command connects the specified virtual hard disk, stored on the physical host at L:\OS.VHD, to the first slot (0) of the primary channel (0) on the virtual IDE controller on the virtual machine instead of the default virtual hard disk in the template. The virtual hard disk stored at L:\OS.VHD contains the operating system that will start on the virtual machine. Additionally, this command uses the JobGroup parameter to specify that it will not run until the New-SCVirtualMachine cmdlet triggers the commands in the JobGroup list to run.

The last command triggers all commands that contain the \$JobGroupID variable to run and creates the new virtual machine named VM06 from the template stored in \$VMTemplate. The virtual machine is deployed on the host specified in \$VMHost and the virtual machine is stored in the root directory of the L: drive. The UseLocalVirtualHarddisk parameter specifies that New-SCVirtualMachine use an existing hard disk on the host instead of copying a vhd from the library. Therefore, the virtual hard disk associated with the virtual disk drive on the template is replaced with the virtual hard drive that exists on the host, L:\OS.VHD. As a result, both the virtual machine and its operating system are stored on the L: drive on the host.

The next example stores the virtual machine and its operating system on different drives.

```

PS C:\> $JobGroupID = [Guid]::NewGuid().ToString()
PS C:\> $VMTemplate = Get-SCVMTemplate | where {$_.Name -eq "VMTemplate01"}
PS C:\> $VMHost = Get-SCVMHost -ComputerName "VMHost06.Contoso.com"
PS C:\> Move-SCVirtualHardDisk -IDE -BUS 0 -LUN 0 -Path "L:\OS.VHD" -JobGroup $JobGroupID
PS C:\> New-SCVirtualMachine -Name "VM06" -Path "L:\\" -VMTemplate $VMTemplate -VMHost
$VMHost -JobGroup $JobGroupID -UseLocalVirtualHardDisk

```

---

## 7: Use an existing VHD on the destination host to create a virtual machine from a template, and move another VHD to the new virtual machine.

The first three commands are identical to the first three commands in the previous example. VMTemplate01 is assumed to have a virtual disk drive on IDE Bus 0 and LUN 0 that contains a virtual hard disk.

The fourth command gets the virtual hard disk object named Other.VHD. The Get-SCVirtual HardDisk cmdlet can retrieve virtual hard disk objects from a virtual machine, from a template, or from a standalone file stored in the VMM library. Specifying the -All parameter retrieves a full list of all the subordinate objects independent of the parent object. In this case, the command retrieves all the available virtual hard disk objects and then selects Other.VHD.

The fifth command connects the specified virtual hard disk, stored on the physical host at L:\OS.VHD, to the first slot (0) of the primary channel (0) on the virtual IDE controller on the virtual machine instead of the default virtual hard disk in the template. The virtual hard disk stored at L:\OS.VHD contains the operating system that will start on the virtual machine. Additionally, this command uses the JobGroup parameter to specify that it will not run until the New-SCVirtualMachine cmdlet triggers the commands in the JobGroup list to run.

The sixth command creates a new virtual disk drive object and attaches the virtual hard disk object stored in \$VHD to IDE Bus 0 and LUN 1 on the new drive. The command uses the Path parameter to store the virtual hard disk object in \$VHD in the root directory of the R drive on the virtual machine, and it specifies that its name is Other.VHD. Additionally, this command uses the JobGroup parameter to specify that it will not run until the last command triggers the commands in the JobGroup list to run.

The last command triggers all commands that contain the \$JobGroupID variable to run and creates the new virtual machine named VM07 from the template stored in \$VMTemplate. The virtual machine is deployed on the host specified in \$VMHost and the virtual machine is stored in the D:\VirtualMachinePath folder. The UseLocalVirtualHarddisk parameter specifies that New-SCVirtualMachine use an existing hard disk on the host instead of copying a vhd from the library. Therefore, the virtual hard disk associated with the virtual disk drive on the template is replaced with the virtual hard drive that exists on the host, L:\OS.VHD. As a result, the path to the virtual machine is D:\VirtualMachinePath\VM07, the path to the operating system is L:\OS.VHD, and the path to the other virtual hard disk is R:\Other.VHD.

```
PS C:\> $JobGroupID = [guid]::NewGuid()
PS C:\> $VMTemplate = Get-SCVMTemplate | where {$_.Name -eq "VMTemplate01"}
PS C:\> $VMHost = Get-SCVMHost -ComputerName "VMHost07.Contoso.com"
PS C:\> $VHD = Get-SCVirtualHardDisk -All | where {$_.Name -eq "Other.Vhd"}
PS C:\> Move-SCVirtualHardDisk -IDE -BUS 0 -LUN 0 -Path "L:\OS.VHD" -JobGroup $JobGroupID
PS C:\> New-SCVirtualDiskDrive -VirtualHardDisk $VHD -IDE -BUS 0 -LUN 1 -Path "R:\" -
Filename "Other.Vhd" -JobGroup $JobGroupID
PS C:\> New-SCVirtualMachine -Name "VM07" -Path "D:\VirtualMachinePath" -VMTemplate
$VMTemplate -VMHost $VMHost -JobGroup $JobGroupID -UseLocalVirtualHardDisk
```

---

## 8: Create a Linux-based virtual machine from a virtual machine template and deploy it on a host.

The first command gets the virtual machine template object named CentOSConfigurable and stores the object in the \$Template variable.

The second command gets the host object named VMHost02 and stores the object in the \$VMHost variable.

The third command gets a credential object (user: root and a password) and stored the object in the \$Cred variable.

The last command creates a new virtual machine from the virtual machine template stored in \$Template, names the virtual machine MyCentOSVM, and deploys the virtual machine on host VMHost02, storing its files at C:\VirtualMachinePath. When the virtual machine is created, the following properties are customized: the computer name for the virtual machine (MyCentOSVM), the Linux dnsdomainname (contoso.com) and the root account password (\$cred). Using the RunAsynchronously parameter returns control to the shell immediately, before the job completes.

```
PS C:\> $VMTemplate = Get-SCVMTemplate -VMMServer "VMMServer01.Contoso.com" | where {$_.Name -eq "CentOSConfigurable"}
```

```
PS C:\> $VMHost = Get-SCVMHost -ComputerName "VMHost02.Contoso.com"
```

```
PS C:\> $Cred = Get-Credential
```

```
PS C:\> New-SCVirtualMachine -VMTemplate $t -Name "MyCentOSVM" -RunAsynchronously -
ComputerName "MyCentOSVM" -LinuxDomainName "Contoso.com" -LocalAdministratorCredential $Cred
-VMHost $VMhost -Path "C:\VirtualMachinePath"
```

### Related topics

[Get-SCVirtualMachine](#)

[Move-SCVirtualMachine](#)

[New-SCP2V](#)

[New-SCV2V](#)

[Read-SCVirtualMachine](#)

[Remove-SCVirtualMachine](#)

[Repair-SCVirtualMachine](#)

[Resume-SCVirtualMachine](#)

[Save-SCVirtualMachine](#)

[Set-SCVirtualMachine](#)

[Start-SCVirtualMachine](#)

[Stop-SCVirtualMachine](#)

[Suspend-SCVirtualMachine](#)

---

# New-SCVirtualNetwork

---

## New-SCVirtualNetwork

Creates a virtual network on a host managed by VMM over which virtual machines on that host can communicate.

### Syntax

Parameter Set: NewCluster

```
New-SCVirtualNetwork [-Name] <String> -JobGroup <Guid> -LogicalNetwork <LogicalNetwork[]> [-BoundToVMHost <Boolean> ] [-Description <String> ] [-HostBoundVlanId <UInt16> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

Parameter Set: Cluster

```
New-SCVirtualNetwork [-Name] <String> -LogicalNetwork <LogicalNetwork[]> -VMHostCluster <HostCluster> [-BoundToVMHost <Boolean> ] [-Description <String> ] [-HostBoundVlanId <UInt16> ] [-JobGroup <Guid> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

Parameter Set: Host

```
New-SCVirtualNetwork [-Name] <String> -VMHost <Host> [-BoundToVMHost <Boolean> ] [-Description <String> ] [-HostBoundVlanId <UInt16> ] [-JobGroup <Guid> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMHostNetworkAdapters <HostNetworkAdapter[]> ] [ <CommonParameters>]
```

Parameter Set: LogicalSwitch

```
New-SCVirtualNetwork -LogicalSwitch <LogicalSwitch> -VMHost <Host> [-CreateManagementAdapter] [-Description <String> ] [-JobGroup <Guid> ] [-JobVariable <String> ] [-ManagementAdapterName <String> ] [-ManagementAdapterPortClassification <PortClassification> ] [-ManagementAdapterVlanId <UInt16> ] [-ManagementAdapterVMNetwork <VMNetwork> ] [-ManagementAdapterVMSubnet <VMSubnet> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMHostNetworkAdapters <HostNetworkAdapter[]> ] [ <CommonParameters>]
```

## Detailed Description

The New-SCVirtualNetwork cmdlet creates a virtual network on a host managed by Virtual Machine Manager (VMM) over which virtual machines on that host can communicate.

System Center 2012 - VMM supports the use of virtual switches to implement virtual networking scenarios for Hyper-V and Citrix XenServer hosts. You can connect, or bind, virtual machines and hosts to a virtual network switch in a manner similar to the way that you connect physical computers to a physical network switch.

---

For Hyper-V hosts, and the virtual machines deployed on these hosts, VMM also supports the use of virtual switches to implement virtual local area networks (VLANs). A VLAN is an independent logical virtual network configured within a physical LAN. If you create multiple VLANs on a physical LAN, these separate logical segments cannot exchange data with each other.

For XenServer hosts, all virtual switches attached to a single network adapter on a XenServer host are represented as a single virtual network within VMM.

In VMM for System Center 2012, you can easily move a virtual machine that is connected to a VLAN from one host to another host and (assuming that both hosts are connected to the same VLAN), the virtual machine in its new location is already configured to resume communicating over the VLAN without any additional administrator effort. Moving a virtual machine to a new location on a VLAN does not require software reconfiguration in the way that moving a physical computer to a new location on a physical network requires hardware reconfiguration.

## VMM FOR SYSTEM CENTER 2012 NETWORKING SCENARIOS

-----

The following three scenarios summarize VMM for System Center 2012 virtual networking configurations.

### Scenario 1 – External Virtual Network

In this scenario, virtual machines deployed on a host use a virtual network adapter to connect to a virtual switch on the host, and this virtual switch is, in turn, connected to a physical network adapter on the host. The host is connected through a physical switch to other computers on its network. This configuration gives the virtual machines access to the host itself, to the physical network to which the host is connected, and to other physical computers (or other physical devices) that are on the same physical network as the host.

The virtual network can support external access though a VLAN if the physical adapter on the host that it is bound to has been configured appropriately and if the virtual machines on that host are configured to use a VLAN. For more information, type: "Get-Help Add-SCVMHostNetworkAdapter -detailed", or "Get-Help New-SCVirtualNetworkAdapter -detailed".

### Scenario 2 – Internal Virtual Network

In this scenario, virtual machines deployed on a host use a virtual network adapter to connect to a virtual switch on the host. In this scenario, the virtual network is bound to the host but the virtual machines do not connect via the virtual switch to a physical network adapter on the host. This configuration establishes an internal virtual network that enables virtual machines connected to that virtual switch to communicate with each other and with services and applications on the host, but not with other computers connected to the host's physical network.

If you want to configure an internal network that is separated into two or more VLANs, you must set the VLAN IDs on a virtual network adapter configured on the virtual machine object. For more information, type: "Get-Help New-SCVirtualNetworkAdapter -detailed", "Get-Help Set-SCVirtualNetworkAdapter -detailed", or "Get-Help Set-SCVMHostNetworkAdapter -detailed".

### Scenario 3 - Private Virtual Network

In this scenario, virtual machines deployed on a host use a virtual network adapter to connect to a virtual switch on the host. As in scenario 2, a virtual machine does not connect via that virtual switch to a physical network adapter on the host. Unlike scenario 2, the virtual network is not bound to the host. This configuration establishes a private virtual network that virtual machines on the same host can use to communicate with each other, but, in this case, they cannot communicate with services or applications on the host or with any physical computers connected to the host's physical network.

---

## Parameters

### **-BoundToVMHost<Boolean>**

Indicates whether a virtual network is bound to a host. Binding a virtual network to a host enables network communication to the host.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-CreateManagementAdapter**

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Description<String>**

States a description for the specified object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-HostBoundVlanId<UInt16>**

Assigns a VLAN to the virtual network adapter that was created for the host for the specified virtual network.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-JobGroup<Guid>**

Specifies an identifier for a series of commands that will run as a set just before the final command that includes the same job group identifier runs.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                        |       |
|------------------------|-------|
| Aliases                | none  |
| Required?              | false |
| Position?              | named |
| Default Value          | none  |
| Accept Pipeline Input? | false |

---

|                             |       |
|-----------------------------|-------|
| Accept Wildcard Characters? | false |
|-----------------------------|-------|

### **-LogicalNetwork<LogicalNetwork[]>**

Specifies a logical network. A logical network is a named grouping of IP subnets and VLANs that is used to organize and simplify network assignments.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-LogicalSwitch<LogicalSwitch>**

Specifies a logical switch object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-ManagementAdapterName<String>**

|                        |       |
|------------------------|-------|
| Aliases                | none  |
| Required?              | false |
| Position?              | named |
| Default Value          | none  |
| Accept Pipeline Input? | false |



---

|                             |       |
|-----------------------------|-------|
| Accept Wildcard Characters? | false |
|-----------------------------|-------|

### **-ManagementAdapterPortClassification<PortClassification>**

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-ManagementAdapterVlanId<UInt16>**

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-ManagementAdapterVMNetwork<VMNetwork>**

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-ManagementAdapterVMSubnet<VMSubnet>**

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-Name<String>**

Specifies the name of a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | 1     |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VMHost<Host>**

Specifies a virtual machine host object. VMM supports Hyper-V hosts, VMware ESX hosts, and Citrix XenServer hosts.

For more information about each type of host, type: "Get-Help Add-SCVMHost -detailed". See the examples for a specific cmdlet to determine how that cmdlet uses this parameter.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-VMHostCluster<HostCluster>**

Specifies a VMM host cluster object.

|                        |       |
|------------------------|-------|
| Aliases                | none  |
| Required?              | true  |
| Position?              | named |
| Default Value          | none  |
| Accept Pipeline Input? | false |

|                             |       |
|-----------------------------|-------|
| Accept Wildcard Characters? | false |
|-----------------------------|-------|

## **-VMHostNetworkAdapters<HostNetworkAdapter[]>**

Specifies an array of one or more physical network adapter objects on a host to which virtual machines deployed on that host can connect.

Example format: -VMHostNetworkAdapters \$VMHostNICs

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **VirtualNetwork**

## **Examples**

### **1: Create an external virtual network on a host.**

The first command gets the host object named VMHost01 and stores the object in the \$VMHost variable.

The second command gets the physical host network adapter object named HostLANAdapter01 on VMHost01 and stores the object in the \$HostAdapter variable.

The third command creates a virtual network on VMHost01 named ExternalVirtualNetwork01, and connects the new virtual network to the host network adapter HostLANAdapter01.

This virtual network is an external virtual network. It is attached to the physical network adapter on the host and can therefore access the LAN that the host is attached to as if it were another physical computer on that LAN.

```
PS C:\> $VMHost = Get-SCVMHost -ComputerName "VMHost01.Contoso.com"
```

---

```
PS C:\> $HostAdapter = Get-SCVMHostNetworkAdapter -VMHost $VMHost -Name "HostLANAdapter01"
PS C:\> New-SCVirtualNetwork -Name "ExternalVirtualNetwork01" -VMHost $VMHost -
VMHostNetworkAdapter $HostAdapter
```

## 2: Create an internal host-bound virtual network.

The first command gets the host object VMHost01 and stores the object in the \$VMHost variable.

The second command creates a virtual network on VMHost01, names it InternalVNet01, specifies a description and tag, and binds the virtual network to the physical host.

This virtual network is an internal, host-bound virtual network. Because it is not attached to a physical network adapter on the host, it cannot access networks external to the host. Virtual machines that are connected to this internal virtual network on this host can communicate only with each other. Because the network is bound to the host, network communication from virtual machines to the host is also possible.

```
PS C:\> $VMHost = Get-SCVMHost -ComputerName "VMHost01.Contoso.com"
PS C:\> New-SCVirtualNetwork -VMHost $VMHost -Name "InternalVNet01" -Description "Internal
Host-Bound Virtual Network" -BoundToVMHost $True
```

## 3: Create a private virtual network that is not bound to the host.

The first command gets the host object named VMHost01 and stores the object in the \$VMHost variable.

The second command creates a virtual network on VMHost01 named UnboundVirtualNetwork01.

Because the network is not attached to a physical network adapter on the host, it cannot access networks external to the host. Virtual machines that are connected to this internal virtual network on this host can communicate only with each other. Because the virtual network is not bound to the host, network communication to the host is not possible.

```
PS C:\> $VMHost = Get-SCVMHost -ComputerName "VMHost01.Contoso.com"
PS C:\> New-SCVirtualNetwork -Name "UnboundVirtualNetwork01" -VMHost $VMHost
```

## Related topics

[Add-SCVMHostNetworkAdapter](#)

[Get-SCVirtualNetwork](#)

[New-SCVirtualNetworkAdapter](#)

[Remove-SCVirtualNetwork](#)

[Set-SCVirtualNetwork](#)

[Set-SCVirtualNetworkAdapter](#)

[Set-SCVirtualScsiAdapter](#)

[Set-SCVMHostNetworkAdapter](#)

---

# New-SCVirtualNetworkAdapter

---

## New-SCVirtualNetworkAdapter

Creates a virtual network adapter on a virtual machine, virtual machine template, or hardware profile used in VMM.

### Syntax

Parameter Set: VM

```
New-SCVirtualNetworkAdapter [[-VirtualNetwork] <String> ] -VM <VM> [-  
EnableMACAddressSpoofing <Boolean> ] [-EnableVMNetworkOptimization <Boolean> ] [-  
IPv4AddressType {Dynamic | Static} ] [-IPv6AddressType {Dynamic | Static} ] [-JobVariable  
<String> ] [-LogicalNetwork <LogicalNetwork> ] [-MACAddress <String> ] [-MACAddressType  
<String> ] [-NetworkLocation <String> ] [-NetworkTag <String> ] [-NoConnection] [-  
NoVMNetwork] [-PortClassification <PortClassification> ] [-PROTipID <Guid> ] [-  
RunAsynchronously] [-Synthetic] [-VLANEnabled <Boolean> ] [-VLANID <UInt16> ] [-VMNetwork  
<VMNetwork> ] [-VMSubnet <VMSubnet> ] [-VMwarePortGroup <String> ] [ <CommonParameters>]
```

Parameter Set: HardwareProfile

```
New-SCVirtualNetworkAdapter [[-VirtualNetwork] <String> ] -HardwareProfile <HardwareProfile>  
[-EnableMACAddressSpoofing <Boolean> ] [-EnableVMNetworkOptimization <Boolean> ] [-  
IPv4AddressType {Dynamic | Static} ] [-IPv6AddressType {Dynamic | Static} ] [-JobVariable  
<String> ] [-LogicalNetwork <LogicalNetwork> ] [-MACAddress <String> ] [-MACAddressType  
<String> ] [-NetworkLocation <String> ] [-NetworkTag <String> ] [-NoConnection] [-  
NoVMNetwork] [-PortClassification <PortClassification> ] [-PROTipID <Guid> ] [-  
RunAsynchronously] [-Synthetic] [-VLANEnabled <Boolean> ] [-VLANID <UInt16> ] [-VMNetwork  
<VMNetwork> ] [-VMNetworkServiceSetting <String> ] [-VMSubnet <VMSubnet> ] [-VMwarePortGroup  
<String> ] [ <CommonParameters>]
```

Parameter Set: JobGroup

```
New-SCVirtualNetworkAdapter [[-VirtualNetwork] <String> ] -JobGroup <Guid> [-  
EnableMACAddressSpoofing <Boolean> ] [-EnableVMNetworkOptimization <Boolean> ] [-  
IPv4AddressType {Dynamic | Static} ] [-IPv6AddressType {Dynamic | Static} ] [-JobVariable  
<String> ] [-LogicalNetwork <LogicalNetwork> ] [-LogicalSwitch <LogicalSwitch> ] [-  
MACAddress <String> ] [-MACAddressType <String> ] [-NetworkLocation <String> ] [-NetworkTag  
<String> ] [-NoConnection] [-NoVMNetwork] [-PortClassification <PortClassification> ] [-  
PROTipID <Guid> ] [-RunAsynchronously] [-Synthetic] [-VLANEnabled <Boolean> ] [-VLANID  
<UInt16> ] [-VMHost <Host> ] [-VMMServer <ServerConnection> ] [-VMNetwork <VMNetwork> ] [-  
VMNetworkServiceSetting <String> ] [-VMSubnet <VMSubnet> ] [-VMwarePortGroup <String> ] [ <  
<CommonParameters>]
```

Parameter Set: Template

```
New-SCVirtualNetworkAdapter [[-VirtualNetwork] <String> ] -VMTemplate <Template> [-  
EnableMACAddressSpoofing <Boolean> ] [-EnableVMNetworkOptimization <Boolean> ] [-  
IPv4AddressType {Dynamic | Static} ] [-IPv6AddressType {Dynamic | Static} ] [-JobVariable  
<String> ] [-LogicalNetwork <LogicalNetwork> ] [-MACAddress <String> ] [-MACAddressType  
<String> ] [-NetworkLocation <String> ] [-NetworkTag <String> ] [-NoConnection] [-
```

---

```
NoVMNetwork] [-PortClassification <PortClassification> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-Synthetic] [-VlanEnabled <Boolean> ] [-VlanID <UInt16> ] [-VMNetwork <VMNetwork> ] [-VMNetworkServiceSetting <String> ] [-VMSubnet <VMSubnet> ] [-VMwarePortGroup <String> ] [ <CommonParameters>]
```

Parameter Set: VMHost

```
New-SCVirtualNetworkAdapter -LogicalSwitch <LogicalSwitch> -VMHost <Host> [-IPv4Address <String> ] [-IPv4AddressPool <StaticIPAddressPool> ] [-IPv4AddressType {Dynamic | Static} ] [-IPv6Address <String> ] [-IPv6AddressPool <StaticIPAddressPool> ] [-IPv6AddressType {Dynamic | Static} ] [-JobGroup <Guid> ] [-JobVariable <String> ] [-MACAddress <String> ] [-MACAddressType <String> ] [-Name <String> ] [-PortClassification <PortClassification> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VlanEnabled <Boolean> ] [-VlanID <UInt16> ] [-VMNetwork <VMNetwork> ] [-VMSubnet <VMSubnet> ] [ <CommonParameters>]
```

## Detailed Description

The `New-SCVirtualNetworkAdapter` cmdlet creates a virtual network adapter on a virtual machine, virtual machine template, or hardware profile used to create virtual machines managed by Virtual Machine Manager (VMM).

### NETWORK LOCATION

-----

You can use the `New-SCVirtualNetworkAdapter` cmdlet to specify a network location and connect the virtual network adapter to a virtual network configured on the host when you create the adapter, or you can configure those and other settings later by using the `Set-SCVirtualNetworkAdapter` cmdlet.

### STATIC OR DYNAMIC MAC ADDRESS

-----

Additionally, you can specify whether the virtual network adapter uses a static or dynamic MAC address, and you can specify a static MAC address.

### EMULATED OR SYNTHETIC VIRTUAL NETWORK ADAPTERS

-----

You can use the `New-SCVirtualNetworkAdapter` cmdlet to create an adapter whose type is either emulated (the default) or synthetic.

For virtual machines on any type of host (Hyper-V, VMware, or XenServer), you can configure a virtual network adapter on the virtual machine that emulates a specific physical network adapter.

For virtual machines on Hyper-V hosts, if the guest operating system installed on a virtual machine is a virtualization-aware operating system (for example, Windows Server 2008 or Windows Server 2012, and some versions of Linux), VMM lets you configure a high-performance synthetic virtual network adapter on the virtual machine to communicate with the physical hardware on the host. You must explicitly specify that a virtual network adapter is synthetic by using the `Synthetic` parameter.

### VIRTUAL LOCAL AREA NETWORK

-----

VMM includes support for configuring one or more virtual area networks (VLANs) on a host for use by virtual machines deployed on that host. You can use the `New-SCVirtualNetworkAdapter` cmdlet (or the

---

Set-SCVirtualNetworkAdapter cmdlet) with the VLAN parameters to attach the virtual network adapter on a virtual machine to a VLAN. To configure corresponding VLAN settings on the host network adapter, use the Add-SCVMHostNetworkAdapter cmdlet or the Set-SCVMHostNetworkAdapter cmdlet. For an illustration of how to configure VLANs, see the examples for this cmdlet, and see the examples for New-SCVMHostNetworkAdapter and Set-SCVMHostNetworkAdapter.

## Parameters

### **-EnableMACAddressSpoofing<Boolean>**

Enables, when set to \$True, MAC Address spoofing.

|                             |                             |
|-----------------------------|-----------------------------|
| Aliases                     | MACAddressesSpoofingEnabled |
| Required?                   | false                       |
| Position?                   | named                       |
| Default Value               | none                        |
| Accept Pipeline Input?      | false                       |
| Accept Wildcard Characters? | false                       |

### **-EnableVMNetworkOptimization<Boolean>**

Enables, when set to \$True, virtual machine network optimization. This feature improves network performance for virtual machines with network adapters that support virtual machine queue (VMQ) or TCP Chimney Offload. VMQ enables creating a unique network queue for each virtual network adapter. TCP Chimney Offload enables network traffic processing to be offloaded from the networking stack.

|                             |                              |
|-----------------------------|------------------------------|
| Aliases                     | VMNetworkOptimizationEnabled |
| Required?                   | false                        |
| Position?                   | named                        |
| Default Value               | none                         |
| Accept Pipeline Input?      | false                        |
| Accept Wildcard Characters? | false                        |



---

## **-HardwareProfile<HardwareProfile>**

Specifies a hardware profile object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-IPv4Address<String>**

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-IPv4AddressPool<StaticIPAddressPool>**

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-IPv4AddressType<EthernetAddressType>**

Specifies an IPv4 address type.

---

The acceptable values for this parameter are:

|         |  |
|---------|--|
| Dynamic |  |
| Static  |  |

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-IPv6Address<String>**

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-IPv6AddressPool<StaticIPAddressPool>**

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-IPv6AddressType<EthernetAddressType>**

Specifies an IPv6 address type.

The acceptable values for this parameter are:

|         |  |
|---------|--|
| Dynamic |  |
| Static  |  |

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-JobGroup<Guid>**

Specifies an identifier for a series of commands that will run as a set just before the final command that includes the same job group identifier runs.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | false |

|                             |       |
|-----------------------------|-------|
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-LogicalNetwork<LogicalNetwork>**

Specifies a logical network. A logical network is a named grouping of IP subnets and VLANs that is used to organize and simplify network assignments.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-LogicalSwitch<LogicalSwitch>**

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-MACAddress<String>**

Specifies the Media Access Control (MAC) address, or a set of MAC addresses, for a physical or virtual network adapter on a computer. Valid values are: one or more MAC addresses.

Example format for a single MAC address:

-MACAddress "00-15-5D-B4-DC-00"

Example format for a set of MAC addresses:

---

-MACAddress "00-15-5D-B4-DC-00", "00-1A-A0-E3-75-29"

Example format for a set of MAC addresses:

\$Macs = "00-15-5D-B4-DC-00", "00-1A-A0-E3-75-29"

Set-SCPXEserver -MACAddress \$Macs

NOTE: When used with New-SCPXEserver or Set-SCPXEserver, the MACAddress parameter updates the PXE interfaces from which the SCDM PXE Server listens for and responds to PXE requests

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-MACAddressType<String>**

Specifies the type of MAC address to use for a virtual network adapter. Valid values are: Static, Dynamic.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Name<String>**

|                        |       |
|------------------------|-------|
| Aliases                | none  |
| Required?              | false |
| Position?              | named |
| Default Value          | none  |
| Accept Pipeline Input? | false |

|                             |       |
|-----------------------------|-------|
| Accept Wildcard Characters? | false |
|-----------------------------|-------|

### **-NetworkLocation<String>**

Specifies the network location for a physical network adapter or for a virtual network adapter, or changes the default network location of a host's physical network adapter.

Example formats:

-NetworkLocation \$NetLoc (\$NetLoc might contain "Corp.Contoso.com")

-OverrideNetworkLocation \$TRUE -NetworkLocation "HostNICNewLocation.Contoso.com"

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-NetworkTag<String>**

Specifies a word or phrase to associate with a virtual network adapter that is configured to connect to a specific internal or external network on the host. The NetworkTag identifies all virtual machines with the same NetworkTag as members of the same network. VMM uses a NetworkTag (if one exists) when it evaluates hosts as possible candidates on which to deploy a virtual machine. If the host does not include virtual machines on the network with the same NetworkTag as the virtual machine to be placed, the host receives zero stars in the placement process.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-NoConnection**

Disconnects a virtual network adapter from a virtual network.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-NoVMNetwork**

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-PortClassification<PortClassification>**

Specifies the port classification of the switch to which the virtual network adapter connects.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

---

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-Synthetic**

Specifies that a device, such as a virtual network adapter, on a virtual machine deployed on a Hyper-V host is a high-performance synthetic device. Requires a virtualization-aware guest operating system on the virtual machine.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |



---

## **-VirtualNetwork<String>**

Specifies a virtual network object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | 1     |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VlanEnabled<Boolean>**

Enables a virtual LAN (VLAN) for use by virtual machines on a Hyper-V or Citrix XenServer host.

Example format for a single VLAN: -VLANEnabled -VLANMode "Access" -VLANID 35

Example format for multiple VLANs: -VLANEnabled -VLANMode "Trunk" -VLANTrunkID  
1,2,100,200,1124

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VlanID<UInt16>**

Assigns a numerical identifier in the range 1-4094 to a virtual network adapter on a virtual machine or to a physical network adapter on a virtual machine host.

Configure a VlanID on a Hyper-V, VMware ESX, or Citrix XenServer host:

- On an externally bound physical network adapter when the Vlan mode is Access.

Configure a VlanID on a virtual network adapter of a virtual machine:

- Bound to a physical network adapter on the host, or
- Bound to an internal virtual network on the host.

---

Example format: -VLanEnabled  
-VLanMode "Access" -VLANID 35

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VM<VM>**

Specifies a virtual machine object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-VMHost<Host>**

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

---

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VMNetwork<VMNetwork>**

Specifies a VM network object.

To get a VM network object, use the Get-SCVMNetwork cmdlet.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-VMNetworkServiceSetting<String>**

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

---

## **-VMSubnet<VMSubnet>**

Specifies a VM subnet object.

To obtain a VM subnet object, use the Get-SCVMSubnet cmdlet.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VMTemplate<Template>**

Specifies a VMM template object used to create virtual machines.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-VMwarePortGroup<String>**

Specifies the VMware port group.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **VirtualNetworkAdapter**

## Notes

- Requires a VMM virtual machine object, virtual machine template object, or hardware profile object, which can be retrieved by using the Get-SCVirtualMachine, Get-SCVMTemplate, and Get-SCHardwareProfile cmdlets, respectively.

## Examples

### 1: Create a virtual network adapter on a virtual machine.

The first command gets the virtual machine object named VM01 and stores the object in the \$VM variable.

The second command creates a virtual network adapter on VM01.

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM01"
PS C:\> New-SCVirtualNetworkAdapter -VM $VM
```

### 2: Create a virtual network adapter on a virtual machine template.

The first command gets the virtual machine template object named VMTemplate01 and stores the object in the \$VMTemplate variable.

The second command creates a virtual network adapter on VMTemplate01.

```
PS C:\> $VMTemplate = Get-SCVMTemplate | where { $_.Name -eq "VMTemplate01" }
PS C:\> New-SCVirtualNetworkAdapter -VMTemplate $VMTemplate
```

### 3: Create an emulated virtual network adapter and a synthetic virtual network adapter on a hardware profile.

The first command gets the hardware profile object named NewHWProfile01 from the VMM library and stores the object in the \$HWProfile variable.

The second command creates a virtual network adapter (a "native" or emulated adapter) on NewHWProfile01.

The last command creates a synthetic virtual network adapter on NewHWProfile01.

---

```
PS C:\> $HWProfile = Get-SCHardwareProfile | where { $_.Name -eq "NewHWProfile01" }
PS C:\> New-SCVirtualNetworkAdapter -HardwareProfile $HWProfile
PS C:\> New-SCVirtualNetworkAdapter -HardwareProfile $HWProfile -Synthetic
```

#### **4: Create a virtual network adapter on a virtual machine and assign it a unique MAC address.**

The first command gets the virtual machine object named VM04 and stores the object in the \$VM variable.

The second command creates a virtual network adapter on the virtual machine stored in \$VM (VM04) and stores the object in the \$VNIC variable.

The third command gets the MAC address pool object named MAC Address Pool 01 and stores the object in the \$MACPool variable.

The last command gets the next available MAC address from the address pool stored in \$MACPool, and assigns it to the virtual network adapter stored in \$VNIC.

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM04"
PS C:\> $VNIC = New-SCVirtualNetworkAdapter -VM $VM
PS C:\> $MACPool = Get-SCMACAddressPool -Name "MAC Address Pool 01"
PS C:\> Grant-SCMACAddress -MACAddressPool $MACPool -VirtualNetworkAdapter $VNIC
```

#### **5: Create a virtual network adapter with a static MAC address and a specific VLAN ID.**

The first command gets the virtual machine object named VM05 and stores the object in the \$VM variable.

The second command gets the logical network object named LogicalNetwork01 and stores the object in the \$LogicalNet variable.

The third command gets the virtual network object named ExternalVirtualNetwork01 and stores the object in the \$VirtualNet variable.

The last command creates a new virtual network adapter for VM05, connects the adapter to the logical network stored in \$LogicalNet and the virtual network stored in \$VirtualNet. The command provides a static MAC address for the virtual network adapter, enables VLAN and specifies a VLAN ID of 3.

NOTE: This example assumes that that your host is already connected to a VLAN or, if not, that your host has two network adapters. If your host has a single network adapter, assigning the adapter to a VLAN that is unavailable to the VMM server will prevent VMM from managing the host.

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM05"
PS C:\> $LogicalNet = Get-SCLogicalNetwork -Name "LogicalNetwork01"
PS C:\> $VirtualNet = Get-SCVirtualNetwork -Name "ExternalVirtualNetwork01"
PS C:\> New-SCVirtualNetworkAdapter -VM $VM -LogicalNetwork $LogicalNet -VirtualNetwork $VirtualNet -MACAddress "00-16-D3-CC-00-1A" -MACAddressType "Static" -VLANEnabled $True -VLANId 3
```

---

## Related topics

[Add-SCVMHostNetworkAdapter](#)

[Get-SCHardwareProfile](#)

[Get-SCVirtualMachine](#)

[Get-SCVMMServer](#)

[Get-SCVMTemplate](#)

[New-SCVirtualNetwork](#)

[Remove-SCVirtualNetworkAdapter](#)

[Set-SCVirtualNetworkAdapter](#)

[Set-SCVMHostNetworkAdapter](#)

---

# New-SCVirtualNetworkAdapterNativePortProfile

---

## New-SCVirtualNetworkAdapterNativePortProfile

Creates a virtual network adapter native port profile.

### Syntax

Parameter Set: Default

```
New-SCVirtualNetworkAdapterNativePortProfile [-Name] <String> [-AllowIeeePriorityTagging <Boolean> ] [-AllowMacAddressSpoofing <Boolean> ] [-AllowTeaming <Boolean> ] [-Description <String> ] [-EnableDhcpGuard <Boolean> ] [-EnableIov <Boolean> ] [-EnableIPsecOffload <Boolean> ] [-EnableRouterGuard <Boolean> ] [-EnableVmq <Boolean> ] [-JobVariable <String> ] [-MaximumBandwidth <Int32> ] [-MinimumBandwidthAbsolute <Int32> ] [-MinimumBandwidthWeight <Int32> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters> ]
```

### Detailed Description

The New-SCVirtualNetworkAdapterNativePortProfile cmdlet creates a network adapter native port profile.

### Parameters

#### **-AllowIeeePriorityTagging<Boolean>**

Specifies whether outgoing packets from the virtual machine adapter are allowed to be tagged with IEEE 802.1p priority. When set to \$false, the priority value in the packet is reset to zero.

The default value is \$False.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |



---

### **-AllowMacAddressSpoofing<Boolean>**

Specifies whether MAC address spoofing is enabled. MAC address spoofing allows virtual machines to change the source MAC address in outgoing packets to one that is not assigned to them.

The default value is \$False.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-AllowTeaming<Boolean>**

Specifies whether guest teaming is allowed. Guest teaming allows the virtual network adapter to be teamed with other network adapters that are connected to the same switch.

The default value is \$False.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Description<String>**

States a description for the specified object.

|                        |       |
|------------------------|-------|
| Aliases                | none  |
| Required?              | false |
| Position?              | named |
| Default Value          | none  |
| Accept Pipeline Input? | false |

---

|                             |       |
|-----------------------------|-------|
| Accept Wildcard Characters? | false |
|-----------------------------|-------|

### **-EnableDhcpGuard<Boolean>**

Specifies whether DHCP guard is enabled. DHCP guard drops DHCP server messages from unauthorized virtual machines pretending to be DHCP servers.

The default value is \$False.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-EnableIov<Boolean>**

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-EnableIPsecOffload<Boolean>**

Specifies whether IPsec task offloading is enabled. When sufficient hardware resources are not available, the security associations are not offloaded and are handled in software by the guest operating system. Support from a physical network adapter and the guest operating system is required to offload IPsec tasks.

The default value is \$False.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-EnableRouterGuard<Boolean>**

Specifies whether router advertisement and redirection messages from authorized virtual machines pretending to be routers are dropped.

The default value is \$False.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-EnableVmq<Boolean>**

Specifies whether virtual machine queue (VMQ) is enabled. A physical network adapter is required to enable VMQ.

The default value is \$False.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-MaximumBandwidth<Int32>**

Specifies, in Megabits per second (Mbps), the maximum bandwidth for a virtual network adapter port profile. The default value is 0.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-MinimumBandwidthAbsolute<Int32>**

Specifies, in Megabits per second (Mbps), the minimum bandwidth for a virtual network adapter port profile. The default value is 0.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

### **-MinimumBandwidthWeight<Int32>**

Specifies the minimum bandwidth relative to how much bandwidth the virtual network adapter intends to use compared to other virtual network adapters connected to the same switch. Valid values are 0 to 100. The default value is 0.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Name<String>**

Specifies the name of a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | 1     |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                        |       |
|------------------------|-------|
| Aliases                | none  |
| Required?              | false |
| Position?              | named |
| Default Value          | none  |
| Accept Pipeline Input? | false |

|                             |       |
|-----------------------------|-------|
| Accept Wildcard Characters? | false |
|-----------------------------|-------|

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Examples**

### **1: Create a virtual network adapter native port profile**

This command creates a virtual network adapter native port profile named VirtualNetworkAdapterNativePortProf01 and allows MAC address spoofing.

---

```
PS C:\> New-SCVirtualNetworkAdapterNativePortProfile -Name  
"VirtualNetworkAdapterNativePortProf01" -AllowMacAddressSpoofing $True
```

## Related topics

[Get-SCVirtualNetworkAdapterNativePortProfile](#)

[Remove-SCVirtualNetworkAdapterNativePortProfile](#)

[Set-SCVirtualNetworkAdapterNativePortProfile](#)

---

# New-SCVirtualNetworkAdapterPortProfileSet

---

## New-SCVirtualNetworkAdapterPortProfileSet

Creates a virtual network adapter port profile set.

### Syntax

Parameter Set: Default

```
New-SCVirtualNetworkAdapterPortProfileSet [[-Name] <String> ] [[-Description] <String> ] -  
LogicalSwitch <LogicalSwitch> -PortClassification <PortClassification> [-  
IsDefaultPortProfileSet <Boolean> ] [-JobGroup <Guid> ] [-JobVariable <String> ] [-PROTipID  
<Guid> ] [-RunAsynchronously] [-VirtualNetworkAdapterExtensionPortProfiles  
<VirtualNetworkAdapterExtensionPortProfile[]> ] [-VirtualNetworkAdapterNativePortProfile  
<VirtualNetworkAdapterNativePortProfile> ] [-VMMServer <ServerConnection> ] [  
<CommonParameters>]
```

### Detailed Description

The New-SCVirtualNetworkAdapterPortProfileSet creates a virtual network adapter port profile set.

To create the virtual network adapter port profile set, you must provide a port classification and logical switch. To obtain a port classification object, use the Get-SCPortClassification cmdlet. To obtain a logical switch object, use the Get-SCLogicalSwitch cmdlet.

### Parameters

#### -Description<String>

States a description for the specified object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | 2     |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |



---

## **-IsDefaultPortProfileSet<Boolean>**

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-JobGroup<Guid>**

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-LogicalSwitch<LogicalSwitch>**

Specifies a logical switch object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-Name<String>**

Specifies the name of a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | 1     |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PortClassification<PortClassification>**

Specifies the port classification of the switch to which the virtual network adapter connects.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

-

### **VirtualNetworkAdapterExtensionPortProfiles<VirtualNetworkAdapterExtensionPortProfile[]>**

Specifies an array of virtual network adapter extension port profiles.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

-

## **VirtualNetworkAdapterNativePortProfile<VirtualNetworkAdapterNativePortProfile>**

Specifies a virtual network adapter native port profile object.

To obtain a virtual network adapter native port profile object, use the Get-SCVirtualNetworkAdapterNativePortProfile cmdlet.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

---

## Examples

### 1: Create a virtual network adapter port profile set.

The first command gets the port classification object named PortClass01 and stores the object in the \$PortClass variable.

The second command gets the logical switch object named LogicalSwitch01 and stores the object in the \$LogicalSwitch variable.

The last command creates a virtual network adapter port profile set with the name VirtualNetworkAdapterPortProfileSet01 using PortClass01 and LogicalSwitch01.

```
PS C:\> $PortClass = Get-SCPortClassification -Name "PortClass01"
```

```
PS C:\> $LogSwitch = Get-SCLogicalSwitch -Name "LogicalSwitch01"
```

```
PS C:\> New-SCVirtualNetworkAdapterPortProfileSet -Name "VirtualNetworkAdapterPortProfSet01"  
-PortClassification $PortClass -LogicalSwitch $LogSwitch
```

## Related topics

[Get-SCVirtualNetworkAdapterPortProfileSet](#)

[Remove-SCVirtualNetworkAdapterPortProfileSet](#)

[Set-SCVirtualNetworkAdapterPortProfileSet](#)

---

# New-SCVirtualScsiAdapter

---

## New-SCVirtualScsiAdapter

Creates a virtual SCSI adapter on a virtual machine, virtual machine template, or hardware profile used in VMM.

### Syntax

Parameter Set: VM

```
New-SCVirtualScsiAdapter [[-AdapterID] <Byte> ] -VM <VM> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-ScsiControllerType {DefaultTypeNoType | LsiLogic | BusLogic | ParaVirtualSCSI | LsiLogicSAS} ] [-ShareVirtualScsiAdapter <Boolean> ] [-Synthetic] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: HardwareProfile

```
New-SCVirtualScsiAdapter [[-AdapterID] <Byte> ] -HardwareProfile <HardwareProfile> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-ScsiControllerType {DefaultTypeNoType | LsiLogic | BusLogic | ParaVirtualSCSI | LsiLogicSAS} ] [-ShareVirtualScsiAdapter <Boolean> ] [-Synthetic] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: JobGroup

```
New-SCVirtualScsiAdapter [[-AdapterID] <Byte> ] -JobGroup <Guid> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-ScsiControllerType {DefaultTypeNoType | LsiLogic | BusLogic | ParaVirtualSCSI | LsiLogicSAS} ] [-ShareVirtualScsiAdapter <Boolean> ] [-Synthetic] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: Template

```
New-SCVirtualScsiAdapter [[-AdapterID] <Byte> ] -VMTemplate <Template> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-ScsiControllerType {DefaultTypeNoType | LsiLogic | BusLogic | ParaVirtualSCSI | LsiLogicSAS} ] [-ShareVirtualScsiAdapter <Boolean> ] [-Synthetic] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

## Detailed Description

The New-SCVirtualScsiAdapter cmdlet creates a virtual SCSI adapter on a virtual machine, virtual machine template, or hardware profile used in a Virtual Machine Manager (VMM) environment. After you create the virtual SCSI adapter, you can use the Set-SCVirtualScsiAdapter cmdlet to modify its settings.

Note: Using the ShareVirtualScsiAdapter parameter to share a virtual SCSI adapter on a virtual machine in order to enable guest clustering is supported only if the virtual machine is deployed on an

---

ESX host. The SharedVirtualScsiAdapter parameter is not used for a virtual machine on a Hyper-V host because a virtual machine on a Hyper-V host uses iSCSI for shared storage.

A virtual machine on a Citrix XenServer host always has one virtual SCSI adapter. You cannot remove this adapter or add additional adapters.

## Parameters

### **-AdapterID<Byte>**

Specifies the logical unit number, or LUN ID. Hyper-V and XenServer do not expose this value, and it cannot be changed. For a VMware ESX host, the default is 7 and cannot be changed.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-HardwareProfile<HardwareProfile>**

Specifies a hardware profile object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-JobGroup<Guid>**

Specifies an identifier for a series of commands that will run as a set just before the final command that includes the same job group identifier runs.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

|                             |       |
|-----------------------------|-------|
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|



|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-ScsiControllerType<VMSCSIControllerType>**

Specifies a SCSI controller type. Valid values are: DefaultType, NoType, LsiLogic, BusLogic, ParaVirtualSCSI, LsiLogicSAS.

The acceptable values for this parameter are:

|                   |  |
|-------------------|--|
| DefaultTypeNoType |  |
| LsiLogic          |  |
| BusLogic          |  |
| ParaVirtualSCSI   |  |
| LsiLogicSAS       |  |

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-ShareVirtualScsiAdapter<Boolean>**

Specifies that a virtual SCSI adapter will be shared so that it can be used in guest clustering.

TYPE OF HOST      USES THIS PARAMETER

-----      -----

Hyper-V host      No (for guest clustering, use iSCSI storage)

XenServer host      No (Xen VMs always have exactly one SCSI adapter)

---

Note: When sharing a SCSI controller on a virtual machine on an ESX host, VMM defaults the SCSI sharing policy on VMware to "physical."

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Synthetic**

Specifies that a device, such as a virtual network adapter, on a virtual machine deployed on a Hyper-V host is a high-performance synthetic device. Requires a virtualization-aware guest operating system on the virtual machine.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VM<VM>**

Specifies a virtual machine object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

---

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-VMTemplate<Template>**

Specifies a VMM template object used to create virtual machines.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **VirtualSCSIAdapter**

## **Notes**

- Requires a VMM virtual machine object, virtual machine template object, or hardware profile object, which can be retrieved by using the Get-SCVirtualMachine, Get-SCVMTemplate, or Get-SCHardwareProfile cmdlets, respectively.

---

## Examples

### 1: Create a virtual SCSI adapter on a virtual machine.

The first command gets the virtual machine object named VM01 and stores the object in the \$VM variable.

The second command creates a virtual SCSI adapter on VM01.

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM01"
```

```
PS C:\> New-SCVirtualScsiAdapter -VM $VM
```

### 2: Create a virtual SCSI adapter on a virtual machine template.

The first command gets the virtual machine template object named VMTemplate01 from the VMM library and stores the object in the \$VMTemplate variable.

The second command creates a virtual SCSI adapter on VMTemplate01.

```
PS C:\> $VMTemplate = Get-SCVMTemplate | where { $_.Name -eq "VMTemplate01" }
```

```
PS C:\> New-SCVirtualScsiAdapter -VMTemplate $VMTemplate
```

### 3: Create a virtual SCSI adapter on a hardware profile.

The first command gets the hardware profile object named NewHWProfile01 from the VMM library and stores the object in the \$HWProfile variable.

The second command creates a virtual SCSI adapter on NewHWProfile01.

```
PS C:\> $HWProfile = Get-SCHardwareProfile | where { $_.Name -eq "NewHWProfile01" }
```

```
PS C:\> New-SCVirtualScsiAdapter -HardwareProfile $HWProfile
```

## Related topics

[Get-SCHardwareProfile](#)

[Get-SCVirtualMachine](#)

[Get-SCVirtualScsiAdapter](#)

[Get-SCVMMServer](#)

[Get-SCVMTemplate](#)

[Remove-SCVirtualScsiAdapter](#)

[Set-SCVirtualScsiAdapter](#)

---

# New-SCVMCheckpoint

---

## New-SCVMCheckpoint

Creates a checkpoint for a virtual machine deployed on a host managed by VMM.

### Syntax

Parameter Set: VM

```
New-SCVMCheckpoint [-VM] <VM> [-Description <String> ] [-JobVariable <String> ] [-Name <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The New-SCVMCheckpoint cmdlet creates a checkpoint for a virtual machine deployed on a host managed by Virtual Machine Manager (VMM). You can use a checkpoint to restore a virtual machine to a previous state.

A typical use is to create a checkpoint before you install an update to the operating system or to an application on the virtual machine so that, if the update fails or adversely affects the virtual machine, you can use the Restore-VMCheckpoint cmdlet to revert the virtual machine to its previous state.

For virtual machines deployed on a Hyper-V host, VMware ESX host, or Citrix XenServer host, VMM creates the checkpoint without stopping the virtual machine, so no interruption in service occurs.

It is important to back up data files on a virtual machine before you restore the virtual machine to a checkpoint. When you restore the virtual machine, user data files on its virtual hard disks are returned to their previous state.

Although checkpoints let you restore a virtual machine to a previous state after a change such as a system or application update, checkpoints do not provide a permanent backup of the operating system, applications, or files. Checkpoints are stored with the virtual machine on the host. Therefore, if the host fails, checkpoints for virtual machines deployed on that host are lost.

To provide data protection for your virtual machines, you can use the Volume Shadow Copy Service (VSS). You can use a backup application such as Data Protection Manager (DPM) to back up virtual machines on any type of host to external storage.

You can grant self-service users permission to create and manage checkpoints for their virtual machines. For more information, type: "Get-Help Set-VMMUserRole -detailed".

---

## Parameters

### **-Description<String>**

States a description for the specified object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Name<String>**

Specifies the name of a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VM<VM>**

Specifies a virtual machine object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

---

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-Confirm**

Prompts you for confirmation before executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |



---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **VMCheckpoint**

## Examples

### 1: Create a virtual machine checkpoint for virtual machines that have the same name but reside on different hosts.

The first command gets the virtual machine objects named VM01 (this example assumes that more than one host contains a virtual machine named VM01), creates a checkpoint for each virtual machine object, and then stores the checkpoint objects in the \$Checkpoints object array.

The second command displays information about each checkpoint object stored in \$Checkpoints to the user.

```
PS C:\> $Checkpoints = Get-SCVirtualMachine -Name "VM01" | New-SCVMCheckpoint
PS C:\> $Checkpoints
```

### 2: Create a virtual machine checkpoint for a virtual machine asynchronously.

This example creates checkpoints in the same manner as Example 1 except that this command uses the RunAsynchronously parameter to return control to the command shell immediately, and uses the JobVariable parameter to track job progress and store a record of the progress in the NewCheckpointJob variable. When you use the JobVariable parameter, you do not use the dollar sign (\$) to create the variable.

The second command displays the contents of \$NewCheckpointJob.

```
PS C:\> Get-SCVirtualMachine -Name "VM02" | New-SCVMCheckpoint -RunAsynchronously -
JobVariable "NewCheckpointJob"
PS C:\> Write-Host $NewCheckpointJob
```

## Related topics

[Get-SCVMCheckpoint](#)

[Remove-SCVMCheckpoint](#)

[Restore-SCVMCheckpoint](#)

[Set-SCVMCheckpoint](#)

---

# New-SCVMConfiguration

---

## New-SCVMConfiguration

Creates a virtual machine configuration from a virtual machine template.

### Syntax

Parameter Set: FromTemplateOnly

```
New-SCVMConfiguration -VMTemplate <Template> [-AvailabilitySetNames <List<String>> ] [-CostCenter <String> ] [-Description <String> ] [-JobVariable <String> ] [-Name <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

Parameter Set: Cloud

```
New-SCVMConfiguration -Cloud <Cloud> -Name <String> -VMTemplate <Template> [-AvailabilitySetNames <List<String>> ] [-CapabilityProfile <CapabilityProfile> ] [-CostCenter <String> ] [-Description <String> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

Parameter Set: ComputerTier

```
New-SCVMConfiguration -ComputerTier <ComputerTier> [-AvailabilitySetNames <List<String>> ] [-ComputerName <String> ] [-CostCenter <String> ] [-Description <String> ] [-JobVariable <String> ] [-Name <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

Parameter Set: VMHostGroup

```
New-SCVMConfiguration -Name <String> -VMHostGroup <HostGroup> -VMTemplate <Template> [-AvailabilitySetNames <List<String>> ] [-CostCenter <String> ] [-Description <String> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

### Detailed Description

The New-SCVMConfiguration cmdlet creates a virtual machine configuration from a virtual machine template. The virtual machine configuration is used to specify instance-specific values to use when deploying the virtual machine configuration.

### Parameters

#### **-AvailabilitySetNames<List<String>>**

Specifies a list of availability set names.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-CapabilityProfile<CapabilityProfile>**

Specifies a capability profile object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Cloud<Cloud>**

Specifies a private cloud object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-ComputerName<String>**

Specifies the name of a computer that VMM can uniquely identify on your network. Valid formats are: FQDN, IPv4 or IPv6 address, or NetBIOS name.

---

NOTE: See the examples for a specific cmdlet to determine how that cmdlet specifies the computer name.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-ComputerTier<ComputerTier>**

Specifies a computer tier object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-CostCenter<String>**

Specifies the cost center for a virtual machine so that you can collect data about the allocation of virtual machines (or resources allocated to virtual machines) to make use of in your billing system.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-Description<String>**

States a description for the specified object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-Name<String>**

Specifies the name of a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VMHostGroup<HostGroup>**

Specifies a virtual machine host group object or an array of host group objects.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-VMTemplate<Template>**

Specifies a VMM template object used to create virtual machines.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **VMConfiguration**

## **Notes**

- Requires a VMM virtual machine template object, which can be obtained by using the Get-SCVMTemplate cmdlet.

## **Examples**

### **1: Create a virtual machine configuration for placement of a virtual machine on a host group.**

The first command gets the host group object named Production and stores the object in the \$HostGroup variable.

The second command gets all virtual machine template objects, selects the template named VMTemplate01 and then stores the object in the \$VMTemplate variable.

The last command creates a virtual machine configuration named VMConfig01 for the virtual machine template stored in \$VMTemplate for deployment on the host group stored in \$HostGroup, specifying a cost center value of 1234.

```
PS C:\> $HostGroup = Get-SCVMHostGroup -Name "Production"
```

```
PS C:\> $VMTemplate = Get-SCVMTemplate | where {$_.Name -eq "VMTemplate01"}
```

---

```
PS C:\> $VMConfig = New-SCVMConfiguration -VMTemplate $VMTemplate -VMHostGroup $HostGroup -  
CostCenter 1234 -Name "VMConfig01"
```

## Related topics

[Get-SCVMConfiguration](#)

[Get-SCVMTemplate](#)

[Remove-SCVMConfiguration](#)

[Set-SCVMConfiguration](#)

[Update-SCVMConfiguration](#)



---

# New-SCVMHost

---

## New-SCVMHost

Creates a VMM host from a physical computer by using the properties defined in a host profile.

### Syntax

Parameter Set: LegacySkipBMCPowerControl

```
New-SCVMHost -ComputerName <String> -SMBiosGuid <Guid> -VMHostProfile <VMHostProfile> [-BypassADMACHINEAccountCheck] [-Description <String> ] [-IPAddress <String> ] [-JobVariable <String> ] [-LogicalNetwork <LogicalNetwork> ] [-ManagementAdapterMACAddress <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-Subnet <String> ] [-VMHostGroup <HostGroup> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: LegacyUseBMCPowerControl

```
New-SCVMHost -BMCAddress <String> -BMCProtocol {None | IPMI | SMASH | Custom} -BMCRunAsAccount <RunAsAccount> -ComputerName <String> -SMBiosGuid <Guid> -VMHostProfile <VMHostProfile> [-BMCCustomConfigurationProvider <ConfigurationProvider> ] [-BMCPort <UInt32> ] [-BypassADMACHINEAccountCheck] [-Description <String> ] [-IPAddress <String> ] [-JobVariable <String> ] [-LogicalNetwork <LogicalNetwork> ] [-ManagementAdapterMACAddress <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-Subnet <String> ] [-VMHostGroup <HostGroup> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: UseHostConfig

```
New-SCVMHost -VMHostConfig <VMHostConfig> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The New-SCVMHost cmdlet creates a Virtual Machine Manager (VMM) host from a physical computer by using the properties defined in a host profile. The physical computer must have an out-of-band controller.

Before you create a host, ensure that a PXE server has been added to VMM, a host profile has been created, and any needed driver files have been added to the library. NOTE: The PXE server you add to VMM must be in the same subnet as the physical computers that you want to convert to managed Hyper-V hosts.

For more information about the types of hosts supported by VMM, type: "Get-Help Add-SCVMHost -detailed".

---

## Parameters

### **-BMCAddress<String>**

Specifies, or updates, the out-of-band baseboard management controller (BMC) address for a specific physical machine. This might be an IP address, the fully qualified domain name (FQDN), or the DNS prefix (which is usually the same name as the NetBIOS name).

Typically, the BMC address and its connection to the network are separate from the IP address associated with a standard network adapter. Alternatively, some computers do use a standard network adapter to provide a single address for the BMC and for the network adapter. However, the BMC address has a unique port and is thus uniquely identifiable on the network.

Example IPv4 format: -BMCAddress "10.0.0.21"

Example Ipv6 format: -BMCAddress "2001:4898:2a:3:657b:9c7a:e1f0:6829"

Example FQDN format: -BMCAddress "Computer01.Contoso.com"

Example NetBIOS format: -BMCAddress "Computer01"

NOTE: By default, VMM uses an IP address or FQDN for the BMCAddress. However, it is also possible to create a Windows PowerShell module that enables you to specify other types of addresses as the BMC address.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-BMCCustomConfigurationProvider<ConfigurationProvider>**

Specifies, or updates, a configuration provider object for a baseboard management controller (BMC). A configuration provider is a plug-in to VMM that translates VMM PowerShell commands to API calls that are specific to a type of baseboard management controller. This parameter should be used with the Custom BMCProtocol.

|                        |       |
|------------------------|-------|
| Aliases                | none  |
| Required?              | false |
| Position?              | named |
| Default Value          | none  |
| Accept Pipeline Input? | false |

|                             |       |
|-----------------------------|-------|
| Accept Wildcard Characters? | false |
|-----------------------------|-------|

### **-BMCPort<UInt32>**

Specifies, or updates, the out-of-band baseboard management controller (BMC) port for a specific physical machine. A BMC port is also known as a service processor port. Example default ports are 623 for IPMI and 443 for SMASH over WS-Man.

Example format: -BMCPort 80

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-BMCProtocol<OutOfBandManagementType>**

Specifies, or updates, the protocol that VMM uses to communicate with the out-of-band baseboard management controller (BMC). Valid values are: IPMI, SMASH, Custom.

A BMC (also known as a service processor or management controller) is a specialized controller on the motherboard of a server that acts an interface between the hardware and system management software. If the motherboard of a physical machine includes a BMC, when the machine is plugged in (whether it is powered off or powered on, and whether or not an operating system is installed), information about system hardware and the state of that system hardware health is available.

Example format: -BMCProtocol "Custom"

NOTE: The Custom protocol requires using the BMCCustomCondigurationProvider.

The acceptable values for this parameter are:

|        |  |
|--------|--|
| None   |  |
| IPMI   |  |
| SMASH  |  |
| Custom |  |

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

|                             |       |
|-----------------------------|-------|
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-BMCRunAsAccount<RunAsAccount>**

Specifies the Run As account to use with the baseboard management controller (BMC) device.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-BypassADMACHINEAccountCheck**

Indicates that New-SCVMhost will reuse a computer account that already exists in Active Directory. By default, New-SCVMHost checks Active Directory for an existing account with the specified name to prevent overwriting computer accounts.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-ComputerName<String>**

Specifies the name of a computer that VMM can uniquely identify on your network. Valid formats are: FQDN, IPv4 or IPv6 address, or NetBIOS name.

---

NOTE: See the examples for a specific cmdlet to determine how that cmdlet specifies the computer name.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Description<String>**

States a description for the specified object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-IPAddress<String>**

Specifies an IPv4 or IPv6 address.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-LogicalNetwork<LogicalNetwork>**

Specifies a logical network. A logical network is a named grouping of IP subnets and VLANs that is used to organize and simplify network assignments.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-ManagementAdapterMACAddress<String>**

Specifies the MAC address of the physical network adapter on the computer that is to be used by the VMM server to communicate with this host.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-SMBiosGuid<Guid>**

Specifies the System Management BIOS globally unique identifier (SMBIOS GUID) for a physical computer that is associated with a record for that physical computer in VMM. SMBIOS defines data structures and access methods that enable a user or application to store and retrieve information about hardware on this computer, such as the name of the system, manufacturer, or the system BIOS version. Windows operating systems retrieve SMBIOS data at system startup and make that data available to programs.

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | true  |
| Position? | named |

|                             |       |
|-----------------------------|-------|
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Subnet<String>**

Specifies an IP subnet (IPv4 or IPv6) in Classless Inter-Domain Routing (CIDR) notation.

Example format for an IPv4 subnet: 192.168.0.1/24

Example format for an IPv6 subnet: FD4A:29CD:184F:3A2C::/64

NOTE: An IP subnet cannot overlap with any other subnet in a host group or child host groups.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMHostConfig<VMHostConfig>**

Specifies a host configuration object.

For information about host configuration objects, see the New-SCVMHostConfig cmdlet.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMHostGroup<HostGroup>**

Specifies a virtual machine host group object.



|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMHostProfile<VMHostProfile>**

Specifies a virtual machine host profile object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

---

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **Host**

## Notes

- Requires a host profile object, which can be retrieved using the Get-SCVMHostProfile cmdlet.

## Examples

### 1: Create a host from a physical computer by using a DHCP-based host profile.

The first command gets the Run As account object named BMCRunAsAcct and stores the object in the \$BMCRAA variable.

The second command discovers the physical computer with the IP address of 10.10.0.1 using the Run As account supplied in \$BMCRAA. It then stores the physical computer in the \$NewPhysicalComputer variable.

The third command gets the host group object named HostGroup01 and stores the object in the \$HostGroup variable.

The fourth command gets the host profile object named HostProfile01 and stores the object in the \$HostProfile variable. HostProfile01 is configured to obtain an IP address through the DHCP service.

The last command creates a host from the physical computer stored in \$NewPhysicalComputer using the host profile stored in \$HostProfile, and gives it the name NewHost01.

```
PS C:\> $BMCRAA = Get-SCRunAsAccount -Name "BMCRunAsAcct"
PS C:\> $NewPhysicalComputer = Find-SCComputer -BMCAddress "10.10.0.1" -BMCRunAsAccount $BMCRAA -BMCProtocol "IPMI"
PS C:\> $HostGroup = Get-SCVMHostGroup -Name "HostGroup01"
PS C:\> $HostProfile = Get-SCVMHostProfile -Name "HostProfile01"
PS C:\> New-SCVMHost -VMHostGroup $HostGroup -VMHostProfile $HostProfile -ComputerName "NewHost01" -BMCAddress $NewPhysicalComputer.BMCAddress -BMCRunAsAccount $BMCRAA -BMCProtocol "IPMI" -SMBIOSGUID $NewPhysicalComputer.SMBIOSGUID -ManagementAdapterMACAddress "\00-1D-D8-B7-1C-00" -LogicalNetwork "LogicalNetwork01" -Subnet "192.168.0.1/24" -IPAddress "192.168.0.91"
```

### 2: Create a host from a physical computer by using a static IP-based host profile.

The first command gets the Run As account object named BMCRunAsAccount and stores the object in the \$BMCRAA variable.

The second command discovers the computer with the address 10.10.0.1, using the Run As account stored in \$BMCRAA, and then stores the computer in the \$NewPhysicalComputer variable.

---

The third command gets the host group object named HostGroup01 and stores the object in the \$HostGroup variable.

The fourth command gets the host profile object named HostProfile02 and stores the object in the \$HostProfile variable. HostProfile02 is configured to allocate a static IP address.

The last command creates a host from the physical computer stored in \$NewPhysicalComputer using the host profile stored in \$HostProfile, gives it the name NewHost02, and configures the MAC address, IPAddress, and subnet.

```
PS C:\> $BMCRAA = Get-SCRunAsAccount -Name "BMCRUnAsAcct"
PS C:\> $NewPhysicalComputer = Find-SCComputer -BMCAddress "10.10.0.1" -BMCRUnAsAccount $BMCRAA -BMCPProtocol "IPMI"
PS C:\> $HostGroup = Get-SCVMHostGroup -Name "HostGroup01"
PS C:\> $HostProfile = Get-SCVMHostProfile -Name "HostProfile02"
PS C:\> New-SCVMHost -VMHostGroup $HostGroup -VMHostProfile $HostProfile -BMCAddress $NewPhysicalComputer.BMCAddress -BMCRUnAsAccount $BMCRAA -BMCPProtocol "IPMI" -SMBIOSGUID $NewPhysicalComputer.SMBIOSGUID -ComputerName "NewHost02" -ManagementAdapterMACAddress "00-18-8B-0A-4D-76" -LogicalNetwork "LogicalNetwork01" -Subnet "192.168.1.1/24" -IPAddress "192.168.1.101"
```

### 3: Redeploy an existing host with a new host profile.

The first command gets the Run As account object named BMCRUnAsAccount and stores the object in the \$BMCRAA variable.

The next eight commands save properties from the host that you are going to redeploy.

The tenth command removes the old host from VMM.

The eleventh command gets the host group object named New HostGroup01 and stores the object in the \$NewHostGroup variable.

The twelfth command gets the host profile object named HostProfile02 and stores the object in the \$NewHostProfile variable.

The last command redeploys the old host using the previous settings that identify the host, but to a new host group (stored in \$NewHostGroup) and with updated profile settings (stored in \$NewHostProfile).

```
PS C:\> $BMCRAA = Get-SCRunAsAccount -Name "BMCRUnAsAcct"
PS C:\> $OldHost = Get-SCVMHost "NewHost02"
PS C:\> $OldBMCIP = $OldHost.physicalmachine.BMCAddress
PS C:\> $OldBMCProtocol = $OldHost.physicalmachine.BMCTYPE
PS C:\> $OldComputer = Find-SCComputer -BMCAddress $OldBMCIP -BMCRUnAsAccount $BMCRAA -BMCPProtocol $OldBMCProtocol
PS C:\> $OldGuid = $OldComputer.SMBIOSGUID
PS C:\> $OldAdapter = Get-SCVMHostNetworkAdapter -VMHost $OldHost
PS C:\> $OldMAC = $OldAdapter[0].macaddress
PS C:\> $OldRAA = Get-SCRunAsAccount -Name "RunAsAccount01"
PS C:\> Remove-SCVMHost $OldHost -credential $OldRAA
PS C:\> $NewHostGroup = Get-SCVMHostGroup -Name "HostGroup01"
PS C:\> $NewHostProfile = Get-SCVMHostProfile -Name "HostProfile02"
```

---

```
PS C:\> New-SCVMHost -VMHostGroup $NewHostGroup -VMHostProfile $NewHostProfile -BMCAddress
$OldBMCIP -BMCRunAsAccount $BMcRAA -BMCProtocol $OldBMCProtocol -SMBIOSGUID $OldGUID -
ManagementAdapterMACAddress $OldMAC -ComputerName "Computer01" -LogicalNetwork
"LogicalNetwork01" -Subnet "192.168.0.1/24" -IPAddress "192.168.0.93"
```

## Related topics

[Add-SCVMHost](#)

[Disable-SCVMHost](#)

[Enable-SCVMHost](#)

[Find-SCComputer](#)

[Get-SCVMHost](#)

[Move-SCVMHost](#)

[Read-SCVMHost](#)

[Register-SCVMHost](#)

[Remove-SCVMHost](#)

[Set-SCVMHost](#)

---

# New-SCVMHostConfig

---

## New-SCVMHostConfig

Creates a host configuration object.

### Syntax

Parameter Set: SkipBMCPowerControl

```
New-SCVMHostConfig -ComputerName <String> -SkipBmcPowerControl -SMBiosGuid <Guid> -  
VMHostNetworkAdapterConfig <VMHostNetworkAdapterConfig[]> -VMHostProfile <VMHostProfile> [-  
BootDiskVolume <String> ] [-BypassADMachAccountCheck] [-Description <String> ] [-  
JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMHostGroup <HostGroup> ]  
[-VMMServer <ServerConnection> ] [ <CommonParameters> ]
```

Parameter Set: UseBMCPowerControl

```
New-SCVMHostConfig -BMCAddress <String> -BMCProtocol {None | IPMI | SMASH | Custom} -  
BMCRunAsAccount <RunAsAccount> -ComputerName <String> -SMBiosGuid <Guid> -  
VMHostNetworkAdapterConfig <VMHostNetworkAdapterConfig[]> -VMHostProfile <VMHostProfile> [-  
BMCCustomConfigurationProvider <ConfigurationProvider> ] [-BMCPort <Int32> ] [-  
BootDiskVolume <String> ] [-BypassADMachAccountCheck] [-Description <String> ] [-  
JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMHostGroup <HostGroup> ]  
[-VMMServer <ServerConnection> ] [ <CommonParameters> ]
```

### Detailed Description

The New-SCVMHostConfig cmdlet creates a host configuration object that is used during the host OS customization process.

### Parameters

#### **-BMCAddress<String>**

Specifies, or updates, the out-of-band baseboard management controller (BMC) address for a specific physical machine. This might be an IP address, the fully qualified domain name (FQDN), or the DNS prefix (which is usually the same name as the NetBIOS name).

Typically, the BMC address and its connection to the network are separate from the IP address associated with a standard network adapter. Alternatively, some computers do use a standard network adapter to provide a single address for the BMC and for the network adapter. However, the BMC address has a unique port and is thus uniquely identifiable on the network.

Example IPv4 format:     -BMCAddress "10.0.0.21"

---

Example Ipv6 format: -BMCAddress "2001:4898:2a:3:657b:9c7a:e1f0:6829"

Example FQDN format: -BMCAddress "Computer01.Contoso.com"

Example NetBIOS format: -BMCAddress "Computer01"

NOTE: By default, VMM uses an IP address or FQDN for the BMCAddress. However, it is also possible to create a Windows PowerShell module that enables you to specify other types of addresses as the BMC address.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-BMCCustomConfigurationProvider<ConfigurationProvider>**

Specifies, or updates, a configuration provider object for a baseboard management controller (BMC). A configuration provider is a plug-in to VMM that translates VMM PowerShell commands to API calls that are specific to a type of baseboard management controller. This parameter should be used with the Custom BMCProtocol.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-BMCPort<Int32>**

Specifies, or updates, the out-of-band baseboard management controller (BMC) port for a specific physical machine. A BMC port is also known as a service processor port. Example default ports are 623 for IPMI and 443 for SMASH over WS-Man.

Example format: -BMCPort 80

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-BMCProtocol<OutOfBandManagementType>**

Specifies, or updates, the protocol that VMM uses to communicate with the out-of-band baseboard management controller (BMC). Valid values are: IPMI, SMASH, Custom.

A BMC (also known as a service processor or management controller) is a specialized controller on the motherboard of a server that acts an interface between the hardware and system management software. If the motherboard of a physical machine includes a BMC, when the machine is plugged in (whether it is powered off or powered on, and whether or not an operating system is installed), information about system hardware and the state of that system hardware health is available.

Example format: -BMCProtocol "Custom"

NOTE: The Custom protocol requires using the BMCCustomCondigurationProvider.

The acceptable values for this parameter are:

|        |  |
|--------|--|
| None   |  |
| IPMI   |  |
| SMASH  |  |
| Custom |  |

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-BMCRunAsAccount<RunAsAccount>**

Specifies the Run As account to use with the baseboard management controller (BMC) device.

---

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-BootDiskVolume<String>**

Specifies a physical hard disk object for a physical machine.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-BypassADMachineAccountCheck**

Indicates that New-SCVMhost will reuse a computer account that already exists in Active Directory. By default, New-SCVMHost checks Active Directory for an existing account with the specified name to prevent overwriting computer accounts.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |



---

## **-ComputerName<String>**

Specifies the name of a computer that VMM can uniquely identify on your network. Valid formats are: FQDN, IPv4 or IPv6 address, or NetBIOS name.

NOTE: See the examples for a specific cmdlet to determine how that cmdlet specifies the computer name.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-Description<String>**

States a description for the specified object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                        |       |
|------------------------|-------|
| Aliases                | none  |
| Required?              | false |
| Position?              | named |
| Default Value          | none  |
| Accept Pipeline Input? | false |

---

|                             |       |
|-----------------------------|-------|
| Accept Wildcard Characters? | false |
|-----------------------------|-------|

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-SkipBmcPowerControl**

Skips BMC information and deploys an operating system without powering the computer on or off by using its BMC interface.

|               |       |
|---------------|-------|
| Aliases       | none  |
| Required?     | true  |
| Position?     | named |
| Default Value | none  |

|                             |       |
|-----------------------------|-------|
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-SMBiosGuid<Guid>**

Specifies the System Management BIOS globally unique identifier (SMBIOS GUID) for a physical computer that is associated with a record for that physical computer in VMM. SMBIOS defines data structures and access methods that enable a user or application to store and retrieve information about hardware on this computer, such as the name of the system, manufacturer, or the system BIOS version. Windows operating systems retrieve SMBIOS data at system startup and make that data available to programs.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMHostGroup<HostGroup>**

Specifies a virtual machine host group object or an array of host group objects.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMHostNetworkAdapterConfig<VMHostNetworkAdapterConfig[]>**

Specifies a host network adapter configuration object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMHostProfile<VMHostProfile>**

Specifies a virtual machine host profile object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

---

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **HostConfig**

## Related topics

[New-SCVMHostNetworkAdapterConfig](#)

---

# New-SCVMHostGroup

---

## New-SCVMHostGroup

Creates a host group that can contain virtual machine host computers, other host groups, or host clusters.

### Syntax

Parameter Set: Default

```
New-SCVMHostGroup [-Name] <String> [-Description <String> ] [-EnableUnencryptedFileTransfer <Boolean> ] [-InheritNetworkSettings <Boolean> ] [-JobVariable <String> ] [-ParentHostGroup <HostGroup> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters> ]
```

### Detailed Description

The New-SCVMHostGroup cmdlet creates a host group object that can contain host computers on which one or more virtual machines are deployed, other host groups, or host clusters.

Virtual Machine Manager (VMM) provides a default parent host group called All Hosts, to which you can add child host groups. A new host group is empty until you move hosts into it or create one or more child host groups under it. Host groups are organized into a hierarchical and customizable tree structure. In the host group tree, the parent of a new host group is either the default root host group (All Hosts) or a user-created host group.

A host group can be a parent container for any of the following:

- A host or set of hosts
- A host group or set of host groups, and hosts within those host groups
- A host cluster, and hosts (nodes) within that host cluster

Hosts contained in a host group have a host path property that shows the location of that host in the host group hierarchy, as illustrated in the following table:

| Name             | Path                                     |
|------------------|--|
| ----             | ----                                     |
| All Hosts        | All Hosts                                |
| ChildHostGroup01 | All Hosts\ChildHostGroup01               |
| ChildHostGroup02 | All Hosts\ChildHostGroup02               |
| New Datacenter   | All Hosts\New Datacenter                 |
| nested1          | All Hosts\New Datacenter\nested01        |
| nested2          | All Hosts\New Datacenter\nested\nested02 |

---

## Parameters

### **-Description<String>**

States a description for the specified object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-EnableUnencryptedFileTransfer<Boolean>**

Indicates, when set to True, that network file transfers do not require encryption. Allowing unencrypted network file transfers can improve performance if neither the source host nor the destination host requires encryption.

Use this parameter to:

- Enable unencrypted file transfers into, or out of, the library.
- Enable unencrypted file transfers into, out of, or within a host group.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-InheritNetworkSettings<Boolean>**

Specifies, when set to True, that the network settings for a host group will have the same values as those specified for its parent.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Name<String>**

Specifies the name of a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | 1     |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-ParentHostGroup<HostGroup>**

Specifies the parent host group that contains one or more hosts, host groups, or host clusters.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|



|                             |                |
|-----------------------------|----------------|
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

---

|                             |                |
|-----------------------------|----------------|
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **HostGroup**

## Examples

### 1: Create a host group under the root host group.

This command creates a host group named HostGroup01 on VMMServer01 in the Contoso.com domain. By default, VMM places this host group under the root host group, which is called All Hosts.

```
PS C:\> New-SCVMHostGroup -VMMServer "VMMServer01.Contoso.com" -Name "HostGroup01"
```

### 2: Create a host group under a specified parent host group.

The first command gets the host group named HostGroup01 and stores it in the \$ParentGroup variable. The second command creates a host group named ChildGroup01 and places it under the parent host group stored in the \$ParentGroup variable.

```
PS C:\> $ParentGroup = Get-SCVMHostGroup -Name "HostGroup01"
```

```
PS C:\> New-SCVMHostGroup -Name "ChildGroup01" -ParentHostGroup $ParentGroup
```

## Related topics

[Get-SCVMHostGroup](#)

[Get-SCVMMServer](#)

[Move-SCVMHostGroup](#)

[Remove-SCVMHostGroup](#)

[Set-SCVMHostGroup](#)

---

# New-SCVMHostNetworkAdapterConfig

---

## New-SCVMHostNetworkAdapterConfig

Creates a host network adapter configuration.

### Syntax

Parameter Set: PhysicalGenericNicDhcp

```
New-SCVMHostNetworkAdapterConfig -SetAsGenericNIC -SetAsPhysicalNetworkAdapter -  
UseDhcpForIPConfiguration [-JobVariable <String> ] [-MACAddress <String> ] [-PROTipID <Guid>  
] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: PhysicalGenericNicStaticIP

```
New-SCVMHostNetworkAdapterConfig -LogicalNetwork <LogicalNetwork> -MACAddress <String> -  
SetAsGenericNIC -SetAsPhysicalNetworkAdapter -UseStaticIPForIPConfiguration [-IPv4Address  
<String> ] [-IPv4Subnet <String> ] [-IPv6Address <String> ] [-IPv6Subnet <String> ] [-  
JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer  
<ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: PhysicalManagementNicDhcp

```
New-SCVMHostNetworkAdapterConfig -SetAsManagementNIC -SetAsPhysicalNetworkAdapter -  
UseDhcpForIPConfiguration [-JobVariable <String> ] [-MACAddress <String> ] [-PROTipID <Guid>  
] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: PhysicalManagementNicStaticIP

```
New-SCVMHostNetworkAdapterConfig -LogicalNetwork <LogicalNetwork> -MACAddress <String> -  
SetAsManagementNIC -SetAsPhysicalNetworkAdapter -UseStaticIPForIPConfiguration [-IPv4Address  
<String> ] [-IPv4Subnet <String> ] [-IPv6Address <String> ] [-IPv6Subnet <String> ] [-  
JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer  
<ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: PhysicalNicLogicalSwitch

```
New-SCVMHostNetworkAdapterConfig -LogicalSwitch <LogicalSwitch> -SetAsPhysicalNetworkAdapter  
-UplinkPortProfileSet <UplinkPortProfileSet> [-JobVariable <String> ] [-MACAddress <String>  
] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [  
<CommonParameters>]
```

Parameter Set: VirtualGenericNicDhcp

```
New-SCVMHostNetworkAdapterConfig -LogicalSwitch <LogicalSwitch> -SetAsGenericNIC -  
SetAsVirtualNetworkAdapter -UseDhcpForIPConfiguration -VMNetwork <VMNetwork> [-JobVariable  
<String> ] [-MACAddress <String> ] [-PortClassification <PortClassification> ] [-PROTipID  
<Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: VirtualGenericNicStaticIP

```
New-SCVMHostNetworkAdapterConfig -LogicalSwitch <LogicalSwitch> -SetAsGenericNIC -
```

---

```
SetAsVirtualNetworkAdapter -UseStaticIPForIPConfiguration -VMNetwork <VMNetwork> [-
IPv4Address <String> ] [-IPv4Subnet <String> ] [-IPv6Address <String> ] [-IPv6Subnet
<String> ] [-JobVariable <String> ] [-MACAddress <String> ] [-PortClassification
<PortClassification> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer
<ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: VirtualManagementNicDhcp

```
New-SCVMHostNetworkAdapterConfig -LogicalSwitch <LogicalSwitch> -SetAsManagementNIC -
SetAsVirtualNetworkAdapter -TransientManagementNetworkAdapter <VMHostNetworkAdapterConfig> -
UseDhcpForIPConfiguration -VMNetwork <VMNetwork> [-JobVariable <String> ] [-MACAddress
<String> ] [-PortClassification <PortClassification> ] [-PROTipID <Guid> ] [-
RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: VirtualManagementNicStaticIP

```
New-SCVMHostNetworkAdapterConfig -LogicalSwitch <LogicalSwitch> -SetAsManagementNIC -
SetAsVirtualNetworkAdapter -TransientManagementNetworkAdapter <VMHostNetworkAdapterConfig> -
UseStaticIPForIPConfiguration -VMNetwork <VMNetwork> [-IPv4Address <String> ] [-IPv4Subnet
<String> ] [-IPv6Address <String> ] [-IPv6Subnet <String> ] [-JobVariable <String> ] [-
MACAddress <String> ] [-PortClassification <PortClassification> ] [-PROTipID <Guid> ] [-
RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

## Detailed Description

The New-SCVMHostNetworkAdapterConfig cmdlet creates a host network adapter configuration that is used during the host OS deployment process.

## Parameters

### -IPv4Address<String>

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### -IPv4Subnet<String>

---

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-IPv6Address<String>**

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-IPv6Subnet<String>**

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-LogicalNetwork<LogicalNetwork>**

Specifies a logical network. A logical network is a named grouping of IP subnets and VLANs that is used to organize and simplify network assignments.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-LogicalSwitch<LogicalSwitch>**

Specifies a logical switch object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-MACAddress<String>**

Specifies the Media Access Control (MAC) address, or a set of MAC addresses, for a physical or virtual network adapter on a computer. Valid values are: one or more MAC addresses.

Example format for a single MAC address:

---

-MACAddress "00-15-5D-B4-DC-00"

Example format for a set of MAC addresses:

-MACAddress "00-15-5D-B4-DC-00", "00-1A-A0-E3-75-29"

Example format for a set of MAC addresses:

\$Macs = "00-15-5D-B4-DC-00", "00-1A-A0-E3-75-29"

Set-SCPXEServer -MACAddress \$Macs

NOTE: When used with New-SCPXEServer or Set-SCPXEServer, the MACAddress parameter updates the PXE interfaces from which the SCDM PXE Server listens for and responds to PXE requests

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PortClassification<PortClassification>**

Specifies the port classification of the switch to which the virtual network adapter connects.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | false |
| Position? | named |

---

|                             |       |
|-----------------------------|-------|
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-SetAsGenericNIC**

Indicates that the network adapter is not the management network adapter for the host.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-SetAsManagementNIC**

Indicates that the network adapter is the management network adapter for the host.

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | true  |
| Position? | named |



---

|                             |       |
|-----------------------------|-------|
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-SetAsPhysicalNetworkAdapter**

Indicates that the network adapter is a physical network adapter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-SetAsVirtualNetworkAdapter**

Indicates that the network adapter is a virtual network adapter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

-

### **TransientManagementNetworkAdapter<VMHostNetworkAdapterConfig>**

Specifies the physical network adapter that the management virtual network adapter binds to during the initial operating system deployment phase. This property is used temporarily before the host is deployed.

---

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-UplinkPortProfileSet<UplinkPortProfileSet>**

Specifies an uplink port profile set object.

To obtain an uplink port profile set object, use the Get-SCUplinkPortProfileSet cmdlet.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-UseDhcpForIPConfiguration**

Indicates that the IP address for the network adapter is obtained by using DHCP.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-UseStaticIPForIPConfiguration**

Indicates that the network adapter is configured with a static IP address.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-VMNetwork<VMNetwork>**

Specifies a VM network object.

To get a VM network object, use the Get-SCVMNetwork cmdlet.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **HostNetworkAdapterConfig**

## Examples

### 1: Create a management physical host network adapter configuration by using a static IP address.

The first command gets the logical network object named LogicalNetwork01 and stores the object in the \$LogicalNetwork variable.

The second command creates a physical host network adapter configuration, setting it as the management network adapter and uses a static IP address during IP configuration. The configuration includes LogicalNetwork01, the provided MAC address, and specified subnet.

```
PS C:\> $LogicalNetwork = Get-SCLogicalNetwork -Name "LogicalNetwork01"
```

```
PS C:\> New-SCVMHostNetworkAdapterConfig -SetAsPhysicalNetworkAdapter -SetAsManagementNIC -UseStaticIPForIPConfiguration -LogicalNetwork $LogicalNetwork -MacAddress "00-1D-D8-B7-1C-00" -Subnet "10.0.0.1/24"
```

### 2: Create a physical host network adapter configuration that uses a logical switch.

The first command gets the logical switch object named LogicalSwitch01 and stores the object in the \$LogicalSwitch variable.

The second command gets the uplink port profile set object named UplinkPortProfileSet01 and stores the object in the \$UplinkPortProfSet variable.

The last command creates a physical host network adapter configuration with LogicalSwitch01 and UplinkPortProfileSet01.

```
PS C:\> $LogicalSwitch = Get-SCLogicalSwitch -Name "LogicalSwitch01"
```

```
PS C:\> $UplinkPortProfSet = Get-SCUplinkPortProfileSet -Name "UplinkPortProfileSet01"
```

```
PS C:\> New-SCVMHostNetworkAdapterConfig -SetAsPhysicalNetworkAdapter -LogicalSwitch $LogicalSwitch -UplinkPortProfileSet $UplinkPortProfSet
```

---

### 3: Create a generic virtual host network adapter configuration by using DHCP.

The first command gets the logical switch object named LogicalSwitch01 and stores the object in the \$LogicalSwitch variable.

The second command gets the VM network object named VMNetwork01 and stores the object in the \$VMNetwork variable.

The last command creates a generic virtual host network configuration with LogicalSwitch01 and VMNetwork01 that uses DHCP to obtain an IP address.

```
PS C:\> $LogicalSwitch = Get-SCLogicalSwitch -Name "LogicalSwitch01"
```

```
PS C:\> $VMNetwork = Get-SCVMNetwork -Name "VMNetwork01"
```

```
PS C:\> New-SCVMHostNetworkAdapterConfig -SetAsVirtualNetworkAdapter -SetAsGenericNIC -  
UseDhcpForIPConfiguration -LogicalSwitch $LogicalSwitch -VMNetwork $VMNetwork
```

### Related topics

[Get-SCLogicalSwitch](#)

[Get-SCUplinkPortProfileSet](#)

[New-SCVMHostConfig](#)

---

# New-SCVMHostNetworkAdapterProfile

---

## New-SCVMHostNetworkAdapterProfile

Creates a host network adapter profile.

### Syntax

Parameter Set: PhysicalGenericNicDhcp

```
New-SCVMHostNetworkAdapterProfile -SetAsGenericNIC -SetAsPhysicalNetworkAdapter -  
UseDhcpForIPConfiguration [-ConsistentDeviceName <String> ] [-JobVariable <String> ] [-  
PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [  
<CommonParameters>]
```

Parameter Set: PhysicalGenericNicStaticIP

```
New-SCVMHostNetworkAdapterProfile -LogicalNetwork <LogicalNetwork> -SetAsGenericNIC -  
SetAsPhysicalNetworkAdapter -UseStaticIPForIPConfiguration [-ConsistentDeviceName <String> ]  
[-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer  
<ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: PhysicalManagementNicDhcp

```
New-SCVMHostNetworkAdapterProfile -SetAsManagementNIC -SetAsPhysicalNetworkAdapter -  
UseDhcpForIPConfiguration [-ConsistentDeviceName <String> ] [-JobVariable <String> ] [-  
PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [  
<CommonParameters>]
```

Parameter Set: PhysicalManagementNicStaticIP

```
New-SCVMHostNetworkAdapterProfile -LogicalNetwork <LogicalNetwork> -SetAsManagementNIC -  
SetAsPhysicalNetworkAdapter -UseStaticIPForIPConfiguration [-ConsistentDeviceName <String> ]  
[-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer  
<ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: PhysicalNicLogicalSwitch

```
New-SCVMHostNetworkAdapterProfile -LogicalSwitch <LogicalSwitch> -  
SetAsPhysicalNetworkAdapter -UplinkPortProfileSet <UplinkPortProfileSet> [-  
ConsistentDeviceName <String> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-  
RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: VirtualGenericNicDhcp

```
New-SCVMHostNetworkAdapterProfile -LogicalSwitch <LogicalSwitch> -SetAsGenericNIC -  
SetAsVirtualNetworkAdapter -UseDhcpForIPConfiguration -VMNetwork <VMNetwork> [-JobVariable  
<String> ] [-PortClassification <PortClassification> ] [-PROTipID <Guid> ] [-  
RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: VirtualGenericNicStaticIP

```
New-SCVMHostNetworkAdapterProfile -LogicalSwitch <LogicalSwitch> -SetAsGenericNIC -  
SetAsVirtualNetworkAdapter -UseStaticIPForIPConfiguration -VMNetwork <VMNetwork> [-  
JobVariable <String> ] [-PortClassification <PortClassification> ] [-PROTipID <Guid> ] [-
```

---

RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]

Parameter Set: VirtualManagementNicDhcp

New-SCVMHostNetworkAdapterProfile -LogicalSwitch <LogicalSwitch> -SetAsManagementNIC -  
SetAsVirtualNetworkAdapter -TransientManagementNetworkAdapter <VMHostNetworkAdapterProfile>  
-UseDhcpForIPConfiguration -VMNetwork <VMNetwork> [-JobVariable <String> ] [-  
PortClassification <PortClassification> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-  
VMMServer <ServerConnection> ] [ <CommonParameters>]

Parameter Set: VirtualManagementNicStaticIP

New-SCVMHostNetworkAdapterProfile -LogicalSwitch <LogicalSwitch> -SetAsManagementNIC -  
SetAsVirtualNetworkAdapter -TransientManagementNetworkAdapter <VMHostNetworkAdapterProfile>  
-UseStaticIPForIPConfiguration -VMNetwork <VMNetwork> [-JobVariable <String> ] [-  
PortClassification <PortClassification> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-  
VMMServer <ServerConnection> ] [ <CommonParameters>]

## Detailed Description

The New-SCVMHostNetworkAdapterProfile creates a network adapter profile.

Note that after you have created a network adapter profile, it cannot be updated. To change a network adapter profile, you must remove the network adapter profile and then re-create it.

## Parameters

### **-ConsistentDeviceName<String>**

Specifies a consistent device name for a network adapter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-LogicalNetwork<LogicalNetwork>**

Specifies a logical network. A logical network is a named grouping of IP subnets and VLANs that is used to organize and simplify network assignments.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-LogicalSwitch<LogicalSwitch>**

Specifies a logical switch object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PortClassification<PortClassification>**

Specifies the port classification of the switch to which the virtual network adapter connects.



---

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-SetAsGenericNIC**

Indicates that the network adapter is not the management network adapter for the host.

---

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-SetAsManagementNIC**

Indicates that the network adapter is the management network adapter for the host.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-SetAsPhysicalNetworkAdapter**

Indicates that the network adapter is a physical network adapter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-SetAsVirtualNetworkAdapter**

Indicates that the network adapter is a virtual network adapter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

-

### **TransientManagementNetworkAdapter<VMHostNetworkAdapterProfile>**

Specifies the physical network adapter that the management virtual network adapter binds to during the initial operating system deployment phase. This property is used temporarily before the host is deployed.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-UplinkPortProfileSet<UplinkPortProfileSet>**

Specifies an uplink port profile set object.

To obtain an uplink port profile set object, use the Get-SCUplinkPortProfileSet cmdlet.

|                        |       |
|------------------------|-------|
| Aliases                | none  |
| Required?              | true  |
| Position?              | named |
| Default Value          | none  |
| Accept Pipeline Input? | false |

---

|                             |       |
|-----------------------------|-------|
| Accept Wildcard Characters? | false |
|-----------------------------|-------|

### **-UseDhcpForIPConfiguration**

Indicates that the IP address for the network adapter is obtained by using DHCP.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-UseStaticIPForIPConfiguration**

Indicates that the network adapter is configured with a static IP address.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                        |                |
|------------------------|----------------|
| Aliases                | none           |
| Required?              | false          |
| Position?              | named          |
| Default Value          | none           |
| Accept Pipeline Input? | True (ByValue) |

|                             |       |
|-----------------------------|-------|
| Accept Wildcard Characters? | false |
|-----------------------------|-------|

## **-VMNetwork<VMNetwork>**

Specifies a VM network object.

To get a VM network object, use the Get-SCVMNetwork cmdlet.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **HostNetworkAdapterProfile**

## **Examples**

### **1. Create a generic physical host network adapter profile by using DHCP.**

This command creates a physical host network adapter profile.

The switches used in this command indicate that this is a physical network adapter and that the network adapter is not the management network adapter for the host. Additionally, DHCP is used to obtain an IP address for the network adapter.

```
PS C:\> New-SCVMHostNetworkAdapterProfile -SetAsPhysicalNetworkAdapter -SetAsGenericNIC -
UseDhcpForIPConfiguration -ConsistentDeviceName "Physical host network adapter profile 01"
```

---

## 2: Create a physical network adapter profile with a logical switch.

The first command gets the logical switch object named LogicalSwitch01 and stores the object in the \$LogSwitch variable.

The second command gets the uplink port profile set object named UplinkPortProfileSet01 and stores the object in the \$UplinkPortProfSet variable.

The last command creates a physical host network adapter profile using LogicalSwitch01 and UplinkPortProfileSet01.

```
PS C:\> $LogSwitch = Get-SCLogicalSwitch -Name "LogicalSwitch01"
PS C:\> $UplinkPortProfSet = Get-SCUplinkPortProfileSet -Name "UplinkPortProfileSet01"
PS C:\> New-SCVMHostNetworkAdapterProfile -SetAsPhysicalNetworkAdapter -LogicalSwitch
$LogSwitch -UplinkPortProfileSet $UplinkPortProfSet -ConsistentDeviceName "Physical host
network adapter profile 02"
```

## 3: Create a management virtual host network adapter profile by using a static IP address.

The first command gets the host network adapter profile object with the ID of 259f47c7-c5a9-429d-a421-d232f9b34991 and stores the object in the \$TransMgmtNIC variable.

The second command gets the logical switch object named LogicalSwitch01 and stores the object in the \$LogSwitch variable.

The third command gets the VM network object named VMNetwork01 and stores the object in the \$VMNetwork variable.

The last command creates a virtual host network adapter profile as a management network adapter using LogicalSwitch01 and VMNetwork01.

```
PS C:\> $TransMgmtNIC = Get-SCVMHostNetworkAdapterProfile -ID "259f47c7-c5a9-429d-a421-
d232f9b34991"
PS C:\> $LogSwitch = Get-SCLogicalSwitch -Name "LogicalSwitch01"
PS C:\> $VMNetwork = Get-SCVMNetwork -Name "VMNetwork01"
PS C:\> New-SCVMHostNetworkAdapterProfile -SetAsVirtualNetworkAdapter -SetAsManagementNIC -
TransientManagementNetworkAdapter $TransMgmtNIC -UseStaticIPForIPConfiguration -
LogicalSwitch $LogSwitch -VMNetwork $VMNetwork
```

## Related topics

[Get-SCLogicalSwitch](#)

[Get-SCVMHostNetworkAdapterProfile](#)

[Remove-SCVMHostNetworkAdapterProfile](#)

---

# New-SCVMHostProfile

---

## New-SCVMHostProfile

Creates a host profile.

### Syntax

Parameter Set: CloneHostProfile

```
New-SCVMHostProfile [-Name] <String> -VMHostProfile <VMHostProfile> [-AnswerFile <Script> ]  
[-BypassVHDCConversion <Boolean> ] [-Description <String> ] [-JobVariable <String> ] [-Owner  
<String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [  
<CommonParameters>]
```

Parameter Set: CreateNewHostProfileData

```
New-SCVMHostProfile [-Name] <String> -Domain <String> -DomainJoinRunAsAccount <RunAsAccount>  
-LocalAdministratorCredential <VMCredetial> -VirtualHardDisk <VirtualHardDisk> [-  
AnswerFile <Script> ] [-BypassVHDCConversion <Boolean> ] [-Description <String> ] [-  
DiskConfiguration <String> ] [-DriverMatchingTag <List<String>> ] [-FullName <String> ] [-  
GuiRunOnceCommands <String[]> ] [-JobVariable <String> ] [-OrganizationName <String> ] [-  
Owner <String> ] [-ProductKey <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-TimeZone  
<Int32> ] [-VMHostNetworkAdapterProfile <VMHostNetworkAdapterProfile[]> ] [-VMMServer  
<ServerConnection> ] [-VMPaths <String> ] [ <CommonParameters>]
```

### Detailed Description

The New-SCVMHostProfile cmdlet creates a host profile, Host profiles contain the hardware and operating system configuration settings used to deploy a physical computer as a Hyper-V host.

Before you create a host profile, ensure that the .vhd file that you specify exists in a library share.

### Parameters

#### **-AnswerFile<Script>**

Specifies a script object stored in the VMM library to use as an answer file. The name of the answer file script depends on the operating system that you want to install on a virtual machine:

ANSWER FILE    GUEST OS TO INSTALL ON VM

-----

Sysprep.inf    Windows XP, Windows Server 2000, or Windows Server 2003

Unattend.xml    Windows Vista, Windows 7, or Windows Server 2008

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-BypassVHDCConversion<Boolean>**

Indicates that a dynamic VHD attached to a host profile is not converted to a fixed type during deployment.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Description<String>**

States a description for the specified object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-DiskConfiguration<String>**

Specifies the disk and partition configuration for the physical computer that is to be deployed with Windows Hyper-V.



---

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Domain<String>**

Specifies a fully qualified domain name (FQDN) for an Active Directory domain.

Example format: -Domain "Domain01.Corp.Contoso.com"

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-DomainJoinRunAsAccount<RunAsAccount>**

Specifies a Run As account that has permission to join the specified domain.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-DriverMatchingTag<List<String>>**

Specifies the custom tags to match with driver properties during deployment. When tags are specified, the deployment process matches the tags specified in the host profile with the tags in the drivers in order to download matching drivers and install them in the target image.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-FullName<String>**

Specifies the name of the person in whose name a virtual machine is registered.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-GuiRunOnceCommands<String[]>**

Specifies one or more commands to add to the [GuiRunOnce] section of an unattended answer file (such as Unattend.xml). Use single quotes around each string enclosed in double quotes.

Example format:

```
-GuiRunOnceCommands "C:\APF\APFPostSysPrepCopy.cmd PARAMS1",  
"C:\APF\APFPostSysPrepCopy.cmd PARAMS1"
```

For information about how Windows PowerShell uses quotes, type: "Get-Help about\_Quoting\_Rules"

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | false |

---

|                             |       |
|-----------------------------|-------|
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-LocalAdministratorCredential<VMMSecret>**

Specifies the user name and password for the Local Administrator account (or Linux root account in the case of a Linux compatible Guest Operating System profile).

Specifying credentials on a new or existing template, on a new or existing guest operating system profile, or on a new virtual machine overrides any existing Administrator password.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Name<String>**

Specifies the name of a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | 1     |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-OrganizationName<String>**

Specifies the name of the organization for the person in whose name a virtual machine is registered.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Owner<String>**

Specifies the owner of a VMM object in the form of a valid domain user account.

Example format: -Owner "Contoso\ReneeLo"

Example format: -Owner "ReneeLo@Contoso"

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-ProductKey<String>**

Specifies a product key. The product key is a 25-digit number that identifies the product license. A product key can be used to register VMM or an operating system to be installed on a virtual machine or host.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-TimeZone<Int32>**

Specifies a number (an index) that identifies a geographical region that shares the same standard time. For a list of time zone indexes, see "Microsoft Time Zone Index Values" at: <http://go.microsoft.com/fwlink/?LinkId=120935>. If no time zone is specified, the default time zone used for a virtual machine is the same time zone setting that is on the virtual machine host.

Example format to specify the GMT Standard Time zone: -TimeZone 085

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VirtualHardDisk<VirtualHardDisk>**

Specifies a virtual hard disk object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VMHostNetworkAdapterProfile<VMHostNetworkAdapterProfile[]>**

Specifies a host network adapter profile object.

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | false |
| Position? | named |

|                             |       |
|-----------------------------|-------|
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMHostProfile<VMHostProfile>**

Specifies a virtual machine host profile object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-VMPaths<String>**

Specifies a set of default paths (as strings separated by the pipeline operator) on a host where virtual machine files can be stored.

Example format: -VMPaths "C:\Folder1|C:\Folder2|C:\Folder3"

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

---

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **HostProfile**

## Examples

### 1: Create a host profile.

The first command gets the virtual hard disk object named VHD01.vhd and stores the object in the \$VHD variable.

The second command prompts you for credentials. At the prompt, enter the credentials of an Administrator account. The command stores the credentials in the \$Credential variable.

The third command creates a host profile named HostProfile01 using the virtual hard disk stored in \$VHD, and the credentials stored in \$Credential.

The last command displays information about the newly created host profile to the user.

```
PS C:\> $VHD = Get-SCVirtualHardDisk -Name "VHD01.vhd"
PS C:\> $RunAsAcct = Get-SCRunAsAccount -Name "RunAsAcct01"
PS C:\> $Credential = Get-Credential
PS C:\> $HostProfile = New-SCVMHostProfile -Name "HostProfile01" -Owner "Contoso\Katarina" -
Description "Host Profile 01" -Domain "Contoso.com" -DomainJoinRunAsAccount $RunAsAcct -
VirtualHardDisk $VHD -LocalAdministratorCredential $Credential
PS C:\> $HostProfile
```

### 2: Clone an existing host profile.

The first command gets the host profile object named HostProfile01 and stores the object in the \$OrigHostProfile variable.



---

The second command copies the host profile stored in \$OrigHostProfile, gives it the specified name, and stores the new profile object in the \$NewHostProfile variable.

```
PS C:\> $OrigHostProfile = Get-SCVMHostProfile -Name "HostProfile01"
```

```
PS C:\> $NewHostProfile = New-SCVMHostProfile -Name "HostProfile02" -VMHostProfile  
$OrigHostProfile
```

### 3: Create a host profile that includes a host network adapter profile.

The first command gets the virtual hard disk object named VHDX01.vhdx and stores the object in the \$VHDX variable.

The second command gets the Run As account object named DomainJoinRAA and stores the object in the \$DomainJoinCreds variable.

The third command prompts the user for local administrator credentials and stores the provided credentials in the \$LocalAdminCreds variable.

The fourth command gets the host network adapter profile object with the ID of 259f47c7-c5a9-429d-a421-d232f9b34991 and stores the object in the \$HostNetAdapterProf variable.

The last command creates a host profile named Host Profile 03 using VHDX01 and the host network adapter profile stored in \$HostNetAdapterProf.

```
PS C:\> $VHDX = Get-SCVirtualHardDisk -Name "VHDX01.vhdx"
```

```
PS C:\> $DomainJoinCreds = Get-SCRunAsAccount -Name "DomainJoinRAA"
```

```
PS C:\> $LocalAdminCreds = Get-Credential
```

```
PS C:\> $HostNetAdapterProf = Get-SCVMHostNetworkAdapterProfile -ID "259f47c7-c5a9-429d-  
a421-d232f9b34991"
```

```
PS C:\> New-SCVMHostProfile -Name "HostProfile03" -Owner "Contoso\Katarina" -Description  
"Host Profile 03" -Domain "Contoso.com" -DomainJoinRunAsAccount $DomainJoinCreds -  
VirtualHardDisk $VHDX -LocalAdministratorCredential $LocalAdminCreds -  
VMHostNetworkAdapterProfile $HostNetAdapterProf
```

## Related topics

[Get-SCVMHostProfile](#)

[Remove-SCVMHostProfile](#)

[Set-SCVMHostProfile](#)

---

# New-SCVMNetwork

---

## New-SCVMNetwork

Creates a VM network.

### Syntax

Parameter Set: Default

```
New-SCVMNetwork [-Name] <String> [[-Description] <String> ] -LogicalNetwork <LogicalNetwork> [-CAIPAddressPoolType {IPV4 | IPV6} ] [-JobVariable <String> ] [-Owner <String> ] [-PAIPAddressPoolType {IPV4 | IPV6} ] [-PROTipID <Guid> ] [-RoutingDomainId <Guid> ] [-RunAsynchronously] [-UserRole <UserRole> ] [-VMMServer <ServerConnection> ] [ <CommonParameters> ]
```

Parameter Set: AutoCreate

```
New-SCVMNetwork [-Name] <String> [[-Description] <String> ] -AutoCreateSubnet - LogicalNetwork <LogicalNetwork> [-CAIPAddressPoolType {IPV4 | IPV6} ] [-JobVariable <String> ] [-Owner <String> ] [-PAIPAddressPoolType {IPV4 | IPV6} ] [-PROTipID <Guid> ] [-RoutingDomainId <Guid> ] [-RunAsynchronously] [-UserRole <UserRole> ] [-VMMServer <ServerConnection> ] [ <CommonParameters> ]
```

Parameter Set: External

```
New-SCVMNetwork [-Name] <String> [[-Description] <String> ] -ExternalName <String> - LogicalNetwork <LogicalNetwork> [-CAIPAddressPoolType {IPV4 | IPV6} ] [-JobVariable <String> ] [-Owner <String> ] [-PAIPAddressPoolType {IPV4 | IPV6} ] [-PROTipID <Guid> ] [-RoutingDomainId <Guid> ] [-RunAsynchronously] [-UserRole <UserRole> ] [-VMMServer <ServerConnection> ] [ <CommonParameters> ]
```

Parameter Set: ExternalVMNetwork

```
New-SCVMNetwork [-Name] <String> [[-Description] <String> ] -ExternalVMNetwork - LogicalNetwork <LogicalNetwork> [-CAIPAddressPoolType {IPV4 | IPV6} ] [-JobVariable <String> ] [-Owner <String> ] [-PAIPAddressPoolType {IPV4 | IPV6} ] [-PROTipID <Guid> ] [-RoutingDomainId <Guid> ] [-RunAsynchronously] [-UserRole <UserRole> ] [-VMMServer <ServerConnection> ] [ <CommonParameters> ]
```

Parameter Set: IsolationType

```
New-SCVMNetwork [-Name] <String> [[-Description] <String> ] -IsolationType {NoIsolation | WindowsNetworkVirtualization | External | VLANNetwork} -LogicalNetwork <LogicalNetwork> [-CAIPAddressPoolType {IPV4 | IPV6} ] [-JobVariable <String> ] [-Owner <String> ] [-PAIPAddressPoolType {IPV4 | IPV6} ] [-PROTipID <Guid> ] [-RoutingDomainId <Guid> ] [-RunAsynchronously] [-UserRole <UserRole> ] [-VMMServer <ServerConnection> ] [ <CommonParameters> ]
```

---

## Detailed Description

The New-SCVMNetwork cmdlet creates a VM network. VM networks support multiple methods of isolation: No isolation, network virtualization, external, and VLAN. The type of isolation used by the VM network depends on the specified logical network. Virtual Machine Manager (VMM) uses the IP address pools that are associated with a VM network to assign customer addresses to virtual machines.

The customer address is visible to the virtual machine and is used by customers to communicate with the virtual machine.

You must provide a logical network object when you create a VM network. To obtain a logical network object, use the Get-SCLogicalNetwork cmdlet.

## Parameters

### **-AutoCreateSubnet**

Indicates that a subnet is automatically created.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-CAIPAddressPoolType<VMNetworkIPAddressPoolType>**

Specifies an address pool type. Valid values are: IPV4, IPV6.

The acceptable values for this parameter are:

|      |  |
|------|--|
| IPV4 |  |
| IPV6 |  |

|               |       |
|---------------|-------|
| Aliases       | none  |
| Required?     | false |
| Position?     | named |
| Default Value | none  |

---

|                             |       |
|-----------------------------|-------|
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Description<String>**

States a description for the specified object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | 2     |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-ExternalName<String>**

Specifies an external name for an object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-ExternalVMNetwork**

|               |       |
|---------------|-------|
| Aliases       | none  |
| Required?     | true  |
| Position?     | named |
| Default Value | none  |

|                             |       |
|-----------------------------|-------|
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-IsolationType<VMNetworkType>**

Specifies an isolation type for a VM network. Valid values are: Nolsolation, WindowsNetworkVirtualization, External, VLANNetwork.

The acceptable values for this parameter are:

|                              |  |
|------------------------------|--|
| Nolsolation                  |  |
| WindowsNetworkVirtualization |  |
| External                     |  |
| VLANNetwork                  |  |

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-LogicalNetwork<LogicalNetwork>**

Specifies a logical network. A logical network is a named grouping of IP subnets and VLANs that is used to organize and simplify network assignments.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-Name<String>**

Specifies the name of a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | 1     |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-Owner<String>**

Specifies the owner of a VMM object in the form of a valid domain user account.

Example format: -Owner "Contoso\ReneeLo"

Example format: -Owner "ReneeLo@Contoso"

|                        |       |
|------------------------|-------|
| Aliases                | none  |
| Required?              | false |
| Position?              | named |
| Default Value          | none  |
| Accept Pipeline Input? | false |

|                             |       |
|-----------------------------|-------|
| Accept Wildcard Characters? | false |
|-----------------------------|-------|

### **-PAIPAddressPoolType<VMNetworkIPAddressPoolType>**

Specifies an address pool type. Valid values are: IPV4, IPV6.

The acceptable values for this parameter are:

|      |  |
|------|--|
| IPV4 |  |
| IPV6 |  |

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RoutingDomainId<Guid>**

Specifies the ID of a routing domain in the form of a GUID.

---

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-UserRole<UserRole>**

Specifies a user role object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.



---

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Examples

### 1: Create a VM network.

The first command gets the logical network object named LogicalNetwork01 and stores the object in the \$LogNet variable.

The second command creates a VM network named VMNetwork01 with the logical network LogicalNetwork01.

```
PS C:\> $LogNet = Get-SCLogicalNetwork -Name "LogicalNetwork01"
```

```
PS C:\> New-SCVMNetwork -Name "VMNetwork01" -LogicalNetwork $LogNet
```

## Related topics

[Get-SCVMNetwork](#)

[Remove-SCVMNetwork](#)

[Set-SCVMNetwork](#)

[Get-SCLogicalNetwork](#)

---

# New-SCVMSubnet

---

## New-SCVMSubnet

Creates a VM subnet.

### Syntax

Parameter Set: WindowsNetworkVirtualization

```
New-SCVMSubnet [-Name] <String> [[-Description] <String> ] -SubnetVlan <SubnetVlan[]> -VMNetwork <VMNetwork> [-JobVariable <String> ] [-MaxNumberOfPorts <UInt32> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [-VMSubnetID <UInt32> ] [ <CommonParameters>]
```

Parameter Set: Vlan

```
New-SCVMSubnet [-Name] <String> [[-Description] <String> ] -LogicalNetworkDefinition <LogicalNetworkDefinition> -SubnetVlan <SubnetVlan[]> -VMNetwork <VMNetwork> [-JobVariable <String> ] [-MaxNumberOfPorts <UInt32> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: VMSubnetExternal

```
New-SCVMSubnet [-Name] <String> [[-Description] <String> ] -VMNetwork <VMNetwork> -VMSubnetExternal [-JobVariable <String> ] [-MaxNumberOfPorts <UInt32> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

## Detailed Description

The New-SCVMSubnet cmdlet creates a VM subnet. A VM subnet is an IP subnet that is associated with a VM network.

When you create a VM subnet, you must provide SubNetVlan and VMNetwork objects. To create the SubnetVlan object, use the New-SCSubnetVlan cmdlet. To obtain a VMNetwork object, use the Get-SCVMNetwork cmdlet. The logical network used by the VM network must be enabled for network virtualization.

## Parameters

### **-Description<String>**

States a description for the specified object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | 2     |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-LogicalNetworkDefinition<LogicalNetworkDefinition>**

Specifies a logical network definition (also called a network site) that contains the subnet that the IP address pool serves as specified by the Subnet parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-MaxNumberOfPorts<UInt32>**

Specifies the maximum number of ports supported by the VM subnet.

---

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Name<String>**

Specifies the name of a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | 1     |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-SubnetVlan<SubnetVlan[]>**

Specifies one or more IP subnet and VLAN sets.

For information about creating a SubnetVlan, type: "Get-Help New-SCSubNetVlan".

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

---

## **-VMNetwork<VMNetwork>**

Specifies a VM network object.

To get a VM network object, use the Get-SCVMNetwork cmdlet.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VMSubnetExternal**

Indicates that the VM network is able to communicate with other networks.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VMSubnetID<UInt32>**

Specifies the ID of a VM subnet.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Examples

### 1: Create a VM subnet that uses NVGRE as the encapsulation mechanism.

The first command creates a subnet VLAN object and stores the object in the \$SubNetVLAN variable.

The second command gets the VM network object named VMNetwork 01 and stores the object in the \$VMNetwork variable.

The last command creates a VM subnet with the name VMSubnet01 using the subnet VLAN stored in \$SubnetVLAN and the VM network stored in \$VMNetwork.

```
PS C:\> $SubNetVLAN = New-SCSubnetVLAN -Subnet "192.168.3.0/24"
```

```
PS C:\> $VMNetwork = Get-SCVMNetwork -Name "VMNetwork01"
```

```
PS C:\> New-SCVMSubnet -Name "VMSubnet01" -SubnetVLAN $SubnetVLAN -VMNetwork $VMNetwork
```

## Related topics

[Get-SCVMNetwork](#)

[Get-SCVMSubnet](#)

[New-SCSubnetVLAN](#)

[Remove-SCVMSubnet](#)

[Set-SCVMSubnet](#)

---

# New-SCVMTemplate

---

## New-SCVMTemplate

Creates a virtual machine template used to create virtual machines managed by VMM.

### Syntax

Parameter Set: NewTemplateFromNothing

```
New-SCVMTemplate [-Name] <String> -JobGroup <Guid> [-AnswerFile <Script> ] [-ApplicationProfile <ApplicationProfile> ] [-AutoLogonCount <UInt32> ] [-AutoLogonCredential <RunAsAccount> ] [-CapabilityProfile <CapabilityProfile> ] [-ComputerName <String> ] [-CPUCount <Byte> ] [-CPUMaximumPercent <Int32> ] [-CPUPerVirtualNumaNodeMaximum <Byte> ] [-CPURelativeWeight <Int32> ] [-CPUReserve <UInt16> ] [-CPUType <ProcessorType> ] [-Description <String> ] [-DisableAutoLogon] [-Domain <String> ] [-DomainJoinCredential <VMMSecurityCredential> ] [-DomainJoinOrganizationalUnit <String> ] [-DRProtectionRequired <Boolean> ] [-DynamicMemoryBufferPercentage <Int32> ] [-DynamicMemoryEnabled <Boolean> ] [-DynamicMemoryMaximumMB <Int32> ] [-DynamicMemoryMinimumMB <UInt16> ] [-FullName <String> ] [-GuestOSProfile <GuestOSProfile> ] [-GuiRunOnceCommands <String[]> ] [-HardwareProfile <HardwareProfile> ] [-HAVMPriority <UInt32> ] [-HighlyAvailable <Boolean> ] [-JobVariable <String> ] [-LinuxAdministratorSSHKey <SSHKey> ] [-LinuxDomainName <String> ] [-LocalAdministratorCredential <VMMSecurityCredential> ] [-MemoryMB <Int32> ] [-MemoryPerVirtualNumaNodeMaximumMB <Int32> ] [-MemoryWeight <Int32> ] [-MergeAnswerFile <Boolean> ] [-MonitorMaximumCount <Int32> ] [-MonitorMaximumResolution <String> ] [-NoCustomization] [-NumaIsolationRequired <Boolean> ] [-OperatingSystem <OperatingSystem> ] [-OrganizationName <String> ] [-Owner <String> ] [-ProductKey <String> ] [-PROTipID <Guid> ] [-RemoveDomainJoinOrganizationalUnit] [-RemoveServerFeatures] [-RunAsynchronously] [-SQLProfile <SQLProfile> ] [-TimeZone <Int32> ] [-UnattendSettings <Dictionary<String>> ] [-UserRole <UserRole> ] [-VirtualNumaNodesPerSocketMaximum <Byte> ] [-VirtualVideoAdapterEnabled <Boolean> ] [-VMMServer <ServerConnection> ] [-Workgroup <String> ] [ <CommonParameters> ]
```

Parameter Set: FromTemplate

```
New-SCVMTemplate [-Name] <String> -VMTemplate <Template> [-AnswerFile <Script> ] [-ApplicationProfile <ApplicationProfile> ] [-AutoLogonCount <UInt32> ] [-AutoLogonCredential <RunAsAccount> ] [-CapabilityProfile <CapabilityProfile> ] [-ComputerName <String> ] [-CPUCount <Byte> ] [-CPUMaximumPercent <Int32> ] [-CPUPerVirtualNumaNodeMaximum <Byte> ] [-CPURelativeWeight <Int32> ] [-CPUReserve <UInt16> ] [-CPUType <ProcessorType> ] [-Description <String> ] [-DisableAutoLogon] [-Domain <String> ] [-DomainJoinCredential <VMMSecurityCredential> ] [-DomainJoinOrganizationalUnit <String> ] [-DRProtectionRequired <Boolean> ] [-DynamicMemoryBufferPercentage <Int32> ] [-DynamicMemoryEnabled <Boolean> ] [-DynamicMemoryMaximumMB <Int32> ] [-DynamicMemoryMinimumMB <UInt16> ] [-FullName <String> ] [-GuestOSProfile <GuestOSProfile> ] [-GuiRunOnceCommands <String[]> ] [-HardwareProfile <HardwareProfile> ] [-HAVMPriority <UInt32> ] [-HighlyAvailable <Boolean> ] [-JobGroup <Guid> ] [-JobVariable <String> ] [-LinuxAdministratorSSHKey <SSHKey> ] [-LinuxDomainName <String> ] [-LocalAdministratorCredential <VMMSecurityCredential> ] [-MemoryMB <Int32> ] [-MemoryPerVirtualNumaNodeMaximumMB <Int32> ] [-MemoryWeight <Int32> ] [-MergeAnswerFile <Boolean> ] [-MonitorMaximumCount <Int32> ] [-MonitorMaximumResolution <String> ] [-
```



---

NumaIsolationRequired <Boolean> ] [-OperatingSystem <OperatingSystem> ] [-OrganizationName <String> ] [-Owner <String> ] [-ProductKey <String> ] [-PROTipID <Guid> ] [-RemoveDomainJoinOrganizationalUnit] [-RemoveServerFeatures] [-RunAsynchronously] [-SQLProfile <SQLProfile> ] [-TimeZone <Int32> ] [-UnattendSettings <Dictionary<String>> ] [-UserRole <UserRole> ] [-VirtualNumaNodesPerSocketMaximum <Byte> ] [-VirtualVideoAdapterEnabled <Boolean> ] [-Workgroup <String> ] [ <CommonParameters>]

Parameter Set: FromVHD

New-SCVMTemplate [-Name] <String> -VirtualHardDisk <StandaloneVirtualHardDisk> [-AnswerFile <Script> ] [-ApplicationProfile <ApplicationProfile> ] [-AutoLogonCount <UInt32> ] [-AutoLogonCredential <RunAsAccount> ] [-CapabilityProfile <CapabilityProfile> ] [-ComputerName <String> ] [-CPUCount <Byte> ] [-CPUMaximumPercent <Int32> ] [-CPUPerVirtualNumaNodeMaximum <Byte> ] [-CPURelativeWeight <Int32> ] [-CPUReserve <UInt16> ] [-CPUType <ProcessorType> ] [-Description <String> ] [-DisableAutoLogon] [-Domain <String> ] [-DomainJoinCredential <VMCredial> ] [-DomainJoinOrganizationalUnit <String> ] [-DRProtectionRequired <Boolean> ] [-DynamicMemoryBufferPercentage <Int32> ] [-DynamicMemoryEnabled <Boolean> ] [-DynamicMemoryMaximumMB <Int32> ] [-DynamicMemoryMinimumMB <UInt16> ] [-FullName <String> ] [-GuestOSProfile <GuestOSProfile> ] [-GuiRunOnceCommands <String[]> ] [-HardwareProfile <HardwareProfile> ] [-HAVMPriority <UInt32> ] [-HighlyAvailable <Boolean> ] [-JobGroup <Guid> ] [-JobVariable <String> ] [-LinuxAdministratorSSHKey <SSHKey> ] [-LinuxDomainName <String> ] [-LocalAdministratorCredential <VMCredial> ] [-MemoryMB <Int32> ] [-MemoryPerVirtualNumaNodeMaximumMB <Int32> ] [-MemoryWeight <Int32> ] [-MergeAnswerFile <Boolean> ] [-MonitorMaximumCount <Int32> ] [-MonitorMaximumResolution <String> ] [-NoCustomization] [-NumaIsolationRequired <Boolean> ] [-OperatingSystem <OperatingSystem> ] [-OrganizationName <String> ] [-Owner <String> ] [-ProductKey <String> ] [-PROTipID <Guid> ] [-RemoveDomainJoinOrganizationalUnit] [-RemoveServerFeatures] [-RunAsynchronously] [-SQLProfile <SQLProfile> ] [-TimeZone <Int32> ] [-UnattendSettings <Dictionary<String>> ] [-UserRole <UserRole> ] [-VirtualNumaNodesPerSocketMaximum <Byte> ] [-VirtualVideoAdapterEnabled <Boolean> ] [-Workgroup <String> ] [ <CommonParameters>]

Parameter Set: FromVM

New-SCVMTemplate [-Name] <String> -VM <VM> [-AnswerFile <Script> ] [-ApplicationProfile <ApplicationProfile> ] [-AutoLogonCount <UInt32> ] [-AutoLogonCredential <RunAsAccount> ] [-BootVirtualHardDisk <VirtualHardDisk> ] [-CapabilityProfile <CapabilityProfile> ] [-ComputerName <String> ] [-CPUCount <Byte> ] [-CPUMaximumPercent <Int32> ] [-CPUPerVirtualNumaNodeMaximum <Byte> ] [-CPURelativeWeight <Int32> ] [-CPUReserve <UInt16> ] [-CPUType <ProcessorType> ] [-Description <String> ] [-DisableAutoLogon] [-Domain <String> ] [-DomainJoinCredential <VMCredial> ] [-DomainJoinOrganizationalUnit <String> ] [-DRProtectionRequired <Boolean> ] [-DynamicMemoryBufferPercentage <Int32> ] [-DynamicMemoryEnabled <Boolean> ] [-DynamicMemoryMaximumMB <Int32> ] [-DynamicMemoryMinimumMB <UInt16> ] [-FullName <String> ] [-GuestOSProfile <GuestOSProfile> ] [-GuiRunOnceCommands <String[]> ] [-HardwareProfile <HardwareProfile> ] [-HAVMPriority <UInt32> ] [-HighlyAvailable <Boolean> ] [-JobGroup <Guid> ] [-JobVariable <String> ] [-LibraryServer <LibraryServer> ] [-LinuxAdministratorSSHKey <SSHKey> ] [-LinuxDomainName <String> ] [-LocalAdministratorCredential <VMCredial> ] [-MemoryMB <Int32> ] [-MemoryPerVirtualNumaNodeMaximumMB <Int32> ] [-MemoryWeight <Int32> ] [-MergeAnswerFile <Boolean> ] [-MonitorMaximumCount <Int32> ] [-MonitorMaximumResolution <String> ] [-NoCustomization] [-NumaIsolationRequired <Boolean> ] [-OperatingSystem <OperatingSystem> ] [-OrganizationName <String> ] [-Owner <String> ] [-ProductKey <String> ] [-PROTipID <Guid> ] [-RemoveDomainJoinOrganizationalUnit] [-RemoveServerFeatures] [-RunAsynchronously] [-SharePath <String> ] [-SkipRearm] [-SQLProfile <SQLProfile> ] [-TimeZone <Int32> ] [-

---

```
UnattendSettings <Dictionary<String>> ] [-UserRole <UserRole> ] [-
VirtualNumaNodesPerSocketMaximum <Byte> ] [-VirtualVideoAdapterEnabled <Boolean> ] [-
Workgroup <String> ] [ <CommonParameters>]
```

## Detailed Description

The New-SCVMTemplate cmdlet creates a virtual machine template that administrators or self-service users can use to create virtual machines managed by Virtual Machine Manager (VMM). A template stores hardware configuration and guest operating system information so that a specific template can be used repeatedly to create new virtual machines. The New-SCVMTemplate cmdlet stores the new template object in the VMM library.

You can create a template based on an existing virtual hard disk, an existing template, or from a virtual machine currently deployed on a virtual machine host. If you specify no parameters, VMM creates a default template object.

### VMM TEMPLATE REQUIREMENTS

-----

Following are the requirements for creating a template in VMM:

- **TEMPLATE FROM VHD.** If you create a template from a virtual hard disk, the virtual hard disk must meet the following requirements:
  - **Operating system.** You must install a supported Windows operating system on the virtual hard disk before you use it to create the template. For information about supported operating systems, see System Center Virtual Machine Manager (VMM) 2012 System Requirements and Supported Operating Systems in the TechNet library at <http://go.microsoft.com/fwlink/?LinkID=209608>.
  - **SYSPREP.** You must run the System Preparation tool (Sysprep.exe) on the virtual hard disk to ensure that every copy of the operating system is unique when you distribute it to multiple virtual machines. You can find Sysprep.exe installed on your Windows Server 2008 or later computer at its default location: C:\Windows\System32\sysprep\.
  - **LOCAL ADMIN PASSWORD.** The local Administrator password of the guest operating system on a virtual hard disk that will be used to create a new template must be blank before you run Sysprep.exe on the virtual hard disk. A blank local Administrator password is required to enable you to have the option to specify the local Administrator password when you customize the guest operating system on the template.
- **TEMPLATE FROM VIRTUAL MACHINE.** If you create a template from a virtual machine, the virtual machine will be destroyed during the process of converting it to a template. If you want to keep the virtual machine

---

and also use it to create a template, you can clone the virtual machine before you create the template. For information about how to clone a virtual machine, see `New-SCVirtualMachine`.

- **TEMPLATE FOR SELF-SERVICE USERS.** If a self-service user role includes permission to use a template, the self-service user cannot change any hardware profile settings. The only settings that a self-service user can change when using a template to create a virtual machine are computer name and, if the user has appropriate privileges, password and product ID number.

#### VMM SUPPORT FOR CUSTOMIZABLE OR NON-CUSTOMIZABLE TEMPLATES

-----

If you specify the `NoCustomization` parameter with the `New-SCVMTemplate` cmdlet, you do not need to add a guest operating system profile to the template as you create the template. Without a guest operating system profile, VMM will not require Sysprep to run within the guest operating system when a virtual machine that is created by using this template is deployed on a host.

One possible scenario is that you have a manually sysprepped virtual hard disk that contains a non-Windows operating system (or that contains an operating system that VMM cannot automatically sysprep) and have embedded an answer file in that virtual hard disk that contains the appropriate settings for that operating system. You can use the `New-SCVMTemplate` cmdlet to create a template from this virtual hard disk and specify that the template does not allow customization of the guest operating system. To do so, use the `NoCustomization` parameter when you create the new template.

In another scenario, you might import a VMware-based template that contains a Windows-based operating system into VMM. By default, VMM imports a VMware-based template as customizable if VMM knows how to customize the guest operating system. Otherwise, the template is imported as non-customizable (this is true, for example, for Linux guest operating systems). If you want to create a non-customizable Windows-based template from this imported, customizable VMware-based template, you can use `New-SCVMTemplate` to create a new template, point to the disks that are attached to the imported template, and then specify the `NoCustomization` parameter on the new template.

## Parameters

### **-AnswerFile<Script>**

Specifies a script object stored in the VMM library to use as an answer file. The name of the answer file script depends on the operating system that you want to install on a virtual machine:

ANSWER FILE    GUEST OS TO INSTALL ON VM

-----    -----

Sysprep.inf    Windows XP, Windows Server 2000, or Windows Server 2003

Unattend.xml    Windows Vista, Windows 7, or Windows Server 2008

|         |             |
|---------|-------------|
| Aliases | SysPrepFile |
|---------|-------------|

|                             |                |
|-----------------------------|----------------|
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-ApplicationProfile<ApplicationProfile>**

Specifies an application profile object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-AutoLogonCount<UInt32>**

Specifies the number of times that Windows should automatically log the administrator denoted in the Answer File on to the console session.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-AutoLogonCredential<RunAsAccount>**

Specifies the Run As account contained in the Answer File that Windows uses to log on to the console session when automatic administrator logon is enabled.

---

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-BootVirtualHardDisk<VirtualHardDisk>**

Specifies the virtual hard disk that contains the operating system file to use when you create a new template.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-CapabilityProfile<CapabilityProfile>**

Specifies a capability profile object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-ComputerName<String>**

Specifies the name of a computer that VMM can uniquely identify on your network. Valid formats are: FQDN, IPv4 or IPv6 address, or NetBIOS name.

NOTE: See the examples for a specific cmdlet to determine how that cmdlet specifies the computer name.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-CPUCount<Byte>**

Specifies the number of CPUs on a virtual machine, on a hardware profile, or on a template. See the examples for a specific cmdlet to determine how that cmdlet uses this parameter.

TYPE OF HOST    NUMBER OF PROCESSORS

-----    -----

Hyper-V        Up to 4 CPUs per VM; varies by guest OS

VMware ESX    Up to 4 CPUs per VM for any supported guest OS

Exception: 1 CPU on a VM running Windows NT 4.0

Citrix XenServer Up to 8 CPUs per VM; varies by guest OS

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-CPUMaximumPercent<Int32>**

Specifies the highest percentage of the total resources of a single CPU on the host that can be used by a specific virtual machine at any given time.

Example: -CPUMaximumPercent 80 (to specify 80 per cent)

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-CPUPerVirtualNumaNodeMaximum<Byte>**

Specifies the maximum number of CPUs allowed for each virtual NUMA node.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-CPURelativeWeight<Int32>**

Specifies the amount of CPU resources on a host that this virtual machine can use relative to other virtual machines on the same host. A virtual machine with a higher setting is allocated more CPU resources than a virtual machine with a lower setting.

TYPE OF HOST    RANGE OF RELATIVE VALUES

-----    -----

Hyper-V        1 to 10000

VMware ESX    2000 = High

1500 = Above Normal

1000 = Normal (default)

750 = Below Normal

500 = Low

1 to 1000000 = Custom

The VMware term for these values is "shares."

Citrix XenServer 1 to 65536, normal is 256

NOTE: See the examples for a specific cmdlet to determine how that cmdlet uses this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-CPUReserve<UInt16>**

Specifies the minimum percentage of the resources of a single CPU on the host to allocate to a virtual machine. The percentage of CPU capacity that is available to the virtual machine is never less than this percentage.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-CPUType<ProcessorType>**

Specifies the type of CPU for a virtual machine. To retrieve a list of all CPU types that are available for use in virtual machines in a VMM environment, type: "Get-SCCPUType"

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |



---

## **-Description<String>**

States a description for the specified object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-DisableAutoLogon**

Disables automatic administrator logon.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-Domain<String>**

Specifies a fully qualified domain name (FQDN) for an Active Directory domain.

Example format: -Domain "Domain01.Corp.Contoso.com"

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-DomainJoinCredential<VMMCredential>**

Specifies the user name and password of an account with permission to join a computer to the domain. A limited rights account should be used for joining computers (either physical or virtual) to the domain.

Example format for a PS credential:

```
$DomainJoinCredential = Get-Credential
```

```
-Domain "ThisDomain.Corp.Contoso.com" -DomainJoinCredential $DomainJoinCredential
```

Example format for a Run As account:

```
$DomainJoinCredential = Get-SCRunAsAccount -Name "RunAsAcct01"
```

```
-Domain "ThisDomain.Corp.Contoso.com" -DomainJoinCredential $DomainJoinCredential
```

NOTE: You can use the DomainJoinCredential parameter to specify credentials (on a VMHostProfile) for joining a physical host computer to the domain, or to specify credentials (on a new or existing template, on a new or existing guest operating system profile, or on a new virtual machine) for joining a virtual machine to the domain.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-DomainJoinOrganizationalUnit<String>**

Specifies the organizational unit to which the computer is joined during an unattended mini-setup.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-DRProtectionRequired<Boolean>**

This parameter is reserved for future use.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-DynamicMemoryBufferPercentage<Int32>**

Specifies the percentage of memory above a virtual machine's current memory allocation which the host should try to reserve as a buffer. The default value is 20.

Example format: -DynamicMemoryTargetBufferPercentage 20

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-DynamicMemoryEnabled<Boolean>**

Enables, when set to \$True, dynamic memory for virtual machines. You can enable dynamic memory directly on a virtual machine, or on a template or hardware profile that will be used to create virtual machines. The default value is False.

REQUIRED: You can enable dynamic memory for a virtual machine only if that virtual machine is deployed on a host running Windows Server 2008 SP1 or later or if the virtual machine is stored in a library in a stopped state (hardware changes to a stored virtual machine can only be made if the virtual machine does not have snapshots). Enabling dynamic memory on a virtual machine stored in a library will limit placement of that machine to hosts running Windows Server 2008 SP1 or later.

Example format: -DynamicMemoryEnabled \$True

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | false |

|                             |       |
|-----------------------------|-------|
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-DynamicMemoryMaximumMB<Int32>**

Specifies the maximum amount of memory that can be allocated to a virtual machine if dynamic memory is enabled. The default value is 65536.

REQUIRED: You can enable dynamic memory for a virtual machine only if that virtual machine is deployed on a host running Windows Server 2008 R2 SP1 or later or if the virtual machine is stored in a library in a stopped state (hardware changes to a stored virtual machine can only be made if the virtual machine does not have snapshots). Enabling dynamic memory on a virtual machine stored in a library will limit placement of that machine to hosts running Windows Server 2008 R2 SP1 or later.

Example format: -DynamicMemoryMaximumMB 1024

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-DynamicMemoryMinimumMB<UInt16>**

Specifies the minimum amount of memory that can be allocated to a virtual machine if dynamic memory is enabled. The default value is 65536.

REQUIRED: You can enable dynamic memory for a virtual machine only if that virtual machine is deployed on a host running Windows Server 2008 R2 SP1 or later or if the virtual machine is stored in a library in a stopped state (hardware changes to a stored virtual machine can only be made if the virtual machine does not have snapshots). Enabling dynamic memory on a virtual machine stored in a library will limit placement of that machine to hosts running Windows Server 2008 R2 SP1 or later.

Example format: -DynamicMemoryMinimumMB 1024

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | false |

|                             |       |
|-----------------------------|-------|
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-FullName<String>**

Specifies the name of the person in whose name a virtual machine is registered.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-GuestOSProfile<GuestOSProfile>**

Specifies a guest operating system profile object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-GuiRunOnceCommands<String[]>**

Specifies one or more commands to add to the [GuiRunOnce] section of an unattended answer file (such as Unattend.xml). Use single quotes around each string enclosed in double quotes.

Example format:

```
-GuiRunOnceCommands "C:\APF\APFPostSysPrepCopy.cmd PARAMS1",
"C:\APF\APFPostSysPrepCopy.cmd PARAMS1"
```

---

For information about how Windows PowerShell uses quotes, type: "Get-Help about\_Quoting\_Rules"

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-HardwareProfile<HardwareProfile>**

Specifies a hardware profile object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-HAVMPriority<UInt32>**

Specifies the virtual machine priority. Virtual machines are started in priority order up to the limits of a host cluster node. Valid input for this parameter is a numerical representation of the priority:

3000 = High

2000 = Medium

1000 = Low

0 - Virtual machine does not restart automatically.

Example format: -HAVMPriority 2000

|               |       |
|---------------|-------|
| Aliases       | none  |
| Required?     | false |
| Position?     | named |
| Default Value | none  |

---

|                             |       |
|-----------------------------|-------|
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-HighlyAvailable<Boolean>**

Specifies that a virtual machine will be placed on a Hyper-V host that is part of a host cluster. Configure this setting on a virtual machine, or on a template or hardware profile that will be used to create virtual machines.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobGroup<Guid>**

Specifies an identifier for a series of commands that will run as a set just before the final command that includes the same job group identifier runs.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | false |

---

|                             |       |
|-----------------------------|-------|
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-LibraryServer<LibraryServer>**

Specifies a VMM library server object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-LinuxAdministratorSSHKey<SSHKey>**

Specifies the public key file for a Linux Ssh Key.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-LinuxDomainName<String>**

Specifies a fully qualified domain name (FQDN) to be used in conjunction with Linux operating system specialization.

Example format: -LinuxDomainName "Domain01.Corp.Contoso.com"



|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

**-LocalAdministratorCredential<VMMCredential>**

Specifies the user name and password for the Local Administrator account (or Linux root account in the case of a Linux compatible Guest Operating System profile).

Specifying credentials on a new or existing template, on a new or existing guest operating system profile, or on a new virtual machine overrides any existing Administrator password.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

**-MemoryMB<Int32>**

Specifies, in megabytes (MB), the amount of random access memory (RAM) on the host that is allocated to a virtual machine. The default value is 512 MB. For a virtual machine on which dynamic memory is enabled (on a host running Windows Server 2008 R2 SP1 or later), use MemoryMB to specify the startup memory value.

TYPE OF HOST            MAXIMUM HOST MEMORY ASSIGNABLE TO VM

-----            -----

- Hyper-V                Up to 65536 MB RAM per virtual machine
- VMware ESX Server 3.0.x Up to 16384 MB RAM per virtual machine
- VMware ESX Server 3.5.x Up to 65532 MB RAM per virtual machine
- Citrix XenServer    Up to 32265 MB RAM per VM

Example format: -MemoryMB 1024

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-MemoryPerVirtualNumaNodeMaximumMB<Int32>**

Specifies the maximum amount of memory, in MB, that each virtual NUMA node is allowed.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-MemoryWeight<Int32>**

Indicates the priority in allocating memory to a virtual machine, relative to other virtual machines on the same host. A virtual machine with a higher setting is allocated more memory resources than a virtual machine with a lower setting.

For a host running Windows Server 2008 R2 SP1 or later, 5000 = Normal, 10000 = High, 0 = Low, 1 to 10000 = Custom.

NOTE: See the examples for a specific cmdlet to determine how that cmdlet uses this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-MergeAnswerFile<Boolean>**

Specifies that the cmdlet merge the specified answer file with the specified guest operating system settings. The default value is TRUE. This parameter is used by the VMM console. You do not need to use this parameter at the command prompt.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-MonitorMaximumCount<Int32>**

Specifies the maximum number of monitors supported by a virtual video adapter.

Example format: -MonitorMaximumCount 3

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-MonitorMaximumResolution<String>**

Specifies, as a string, the value that represents the maximum possible monitor resolution of a virtual video adapter. Valid values are: "1024x768", "1280x1024", "1600x1200", "1920x1200". Default value: "1280x1024"

Example format: -MonitorResolutionMaximum "1600x1200"

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | false |
| Position? | named |

---

|                             |       |
|-----------------------------|-------|
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Name<String>**

Specifies the name of a VMM object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-NoCustomization**

Specifies that guest operating system settings on this template cannot be customized.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-NumaIsolationRequired<Boolean>**

Indicates whether NUMA isolation is required.

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | false |
| Position? | named |

|                             |       |
|-----------------------------|-------|
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-OperatingSystem<OperatingSystem>**

Specifies the type of operating system for a virtual machine. To list the names of all available operating systems in VMM, type: "Get-SCOperatingSystem".

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-OrganizationName<String>**

Specifies the name of the organization for the person in whose name a virtual machine is registered.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Owner<String>**

Specifies the owner of a VMM object in the form of a valid domain user account.

Example format: -Owner "Contoso\ReneeLo"

Example format: -Owner "ReneeLo@Contoso"

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-ProductKey<String>**

Specifies a product key. The product key is a 25-digit number that identifies the product license. A product key can be used to register VMM or an operating system to be installed on a virtual machine or host.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RemoveDomainJoinOrganizationalUnit**

Removes the organizational unit that the computer joined during setup.

---

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RemoveServerFeatures**

Removes all server roles and features from a template.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-SharePath<String>**

Specifies a path to a valid library share on an existing library server that uses a Universal Naming Convention (UNC) path.

Example format: `-SharePath "\\LibServer01\LibShare"`

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-SkipRearm**

Skips running the Windows Software Licensing Rearm program. This program restores the Windows operating system to its original, out-of-box licensing state.

**IMPORTANT:** Refer to your licensing agreements with regard to activation and the Rearm program. Using this parameter might violate your licensing agreements.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-SQLProfile<SQLProfile>**

Specifies a SQL Server profile object.

|               |       |
|---------------|-------|
| Aliases       | none  |
| Required?     | false |
| Position?     | named |
| Default Value | none  |



|                             |       |
|-----------------------------|-------|
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-TimeZone<Int32>**

Specifies a number (an index) that identifies a geographical region that shares the same standard time. For a list of time zone indexes, see "Microsoft Time Zone Index Values" at: <http://go.microsoft.com/fwlink/?LinkId=120935>. If no time zone is specified, the default time zone used for a virtual machine is the same time zone setting that is on the virtual machine host.

Example format to specify the GMT Standard Time zone: -TimeZone 085

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-UnattendSettings<Dictionary<String>>**

Specifies a key/value pair consisting of string, string.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-UserRole<UserRole>**

Specifies a user role object.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VirtualHardDisk<StandaloneVirtualHardDisk>**

Specifies a virtual hard disk object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-VirtualNumaNodesPerSocketMaximum<Byte>**

Specifies the maximum number of NUMA nodes allowed for each socket.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VirtualVideoAdapterEnabled<Boolean>**

Enables, when set to \$True, the Microsoft Synthetic 3D Virtual Video Adapter for virtual machines. You can enable the Virtual Video Adapter directly on a virtual machine, or on a template or hardware profile that will be used to create virtual machines.

REQUIRED: You can enable the Microsoft Synthetic 3D Virtual Video Adapter for a virtual machine only if that virtual machine is deployed on a host running Windows Server 2008 R2 SP1 (with the Remote Desktop Services role and Remote Desktop Virtual Graphics role service installed) or later or if the virtual machine is stored in a library in a stopped state (hardware changes to a stored virtual machine can only be made if the virtual machine does not have snapshots). Enabling the Microsoft Synthetic 3D Virtual Video Adapter on a virtual machine stored in a library will limit placement of that machine to hosts running Windows Server 2008 R2 SP1 (with the Remote Desktop Services role and Remote Desktop Virtual Graphics role service installed) or later.

Example format: -VirtualVideoAdapterEnabled \$TRUE

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VM<VM>**

Specifies a virtual machine object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | false |
| Position? | named |

|                             |                |
|-----------------------------|----------------|
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-VMTemplate<Template>**

Specifies a VMM template object used to create virtual machines.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-Workgroup<String>**

Specifies on a new or existing template, on a new or existing guest operating system profile, or on a new virtual machine the name of the workgroup to which you want to join a virtual machine. You can use this parameter to override the existing value on a template or on a guest operating system profile.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

---

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **Template**

## Examples

### 1: Create a virtual machine template from a virtual hard disk.

The first command gets the specified operating system object (64-bit edition of Windows Server 2008 R2 Datacenter) and stores the object in the \$OS variable.

The second command gets the virtual hard disk object named VHD01 from the VMM library on LibServer01 and stores the object in the \$VHD variable.

The last command creates a virtual machine template named VMTemplate01 from VHD01 and specifies the name of the operating system. No customization is made to the operating system.

NOTE: This example assumes that VHD01 is a sysprepped virtual hard disk on which the 64-bit edition of the Windows Server 2008 R2 Datacenter operating system is installed. You can install virtualization guest services on the virtual machine, or VMM will install them automatically when the virtual machine is deployed on a Windows-based host.

```
PS C:\> $OS = Get-SCOperatingSystem -VMMServer "VMMServer01.Contoso.com" | where {$_.Name -eq "64-bit Edition of Windows Server 2008 R2 Datacenter"}
PS C:\> $VHD = Get-SCVirtualHardDisk -VMMServer "VMMServer01.Contoso.com" | where { $_.Name -eq "VHD01.vhd" -and $_.LibraryServer.Name -eq "LibServer01.Contoso.com" }
PS C:\> New-SCVMTemplate -Name "VMTemplate01" -VirtualHardDisk $VHD -OperatingSystem $OS -NoCustomization
```

### 2: Create a virtual machine template from an existing virtual machine.

The first command gets the the library server object named LibServer01 and stores the object in the \$LibraryServer variable.

The second command gets the virtual machine object named VM01 deployed on VMHost02 and stores the object in the \$VM variable.

NOTE: VM01, which is the virtual machine that will be converted to a template, will be destroyed during the conversion process. If you want to retain the virtual machine used to create a template, you can use the New-SCVirtualMachine cmdlet to clone the virtual machine before you create the template.

The third command gets the specified operating system object (Windows Server 2008 R2 Enterprise) and stores the object in the \$OS variable.

The last command creates a virtual machine template named VMTemplate02 from virtual machine VM01. It specifies the owner of the new template, the library server and share where you want to store the new template, and the name of the operating system, without any customization to the operating system. The -RunAsynchronously parameter returns control to the shell immediately before the command completes.

---

```
PS C:\> $LibraryServer = Get-SCLibraryServer | where {$_.Name -eq "LibServer01.Contoso.com"}
PS C:\> $VM = Get-SCVirtualMachine -Name "VM01" | where {$_.VMHost.Name -eq
"VMHost02.Contoso.com"}
PS C:\> $OperatingSystem = Get-SCOperatingSystem | where {$_.Name -eq "64-bit Edition of
Windows Server 2008 R2 Enterprise"}
PS C:\> New-SCVMTemplate -Name "VMTemplate02" -RunAsynchronously -VM $VM -Owner
"Contoso\ReneeLo" -LibraryServer $LibraryServer -SharePath
"\\LibServer01.Contoso.com\MSSCVMMLibrary" -OperatingSystem $OperatingSystem -
NoCustomization
```

### 3: Create a virtual machine template from a virtual hard disk with specified characteristics.

The first command generates a globally unique identifier (GUID) and stores the GUID string in variable \$JobGroupID01. The job group ID functions as an identifier that groups subsequent commands that include \$JobGroupID01 into a single job group.

The second command gets the logical network object named LogicalNetwork01 and stores the object in the \$LogNet variable.

The third command creates a virtual network adapter and uses the JobGroup parameter to indicate that the network adapter is not created until just before the New-SCHardwareProfile cmdlet runs. The New-SCVirtualNetworkAdapter cmdlet sets the MAC address type to dynamic and specifies that the new virtual network adapter will connect to the logical network stored in \$LogNet.

The fourth command creates a virtual SCSI adapter and uses the JobGroup parameter to indicate that the SCSI adapter is not created until just before the New-SCHardwareProfile cmdlet runs. The New-SCVirtualSCSIAdapter cmdlet sets the adapter ID to 6 and sets the Shared parameter to \$False so that the adapter will not be shared, as would be necessary if you wanted to use it in guest clustering.

The fifth command creates a virtual DVD drive and uses the JobGroup parameter to specify that the DVD drive is not created until just before the New-SCHardwareProfile cmdlet runs. The New-SCVirtualDVDDrive cmdlet specifies Bus 1 and LUN 0 to attach the virtual DVD drive to Secondary Channel (0) on the IDE bus.

The sixth command creates a hardware profile named TempHWProfile, sets the owner to Contoso\ReneeLo, specifies a description and that the amount of memory on the host that a virtual machine created by using this template will use is 512 MB. The New-SCHardwareProfile cmdlet uses the JobGroup parameter to specify that all preceding commands that include variable \$JobGroupID01 will run just before New-SCHardwareProfile creates the new hardware profile. After New-SCVirtualNetworkAdapter, New-SCVirtualSCSIAdapter, and New-SCVirtualDVDDrive run, the resulting objects that are created are automatically associated with the new hardware profile.

The seventh command generates a new GUID and stores it in \$JobGroupID02. This job group ID will be used to identify any subsequent commands that include this ID and will delay running those commands until just before the last command that specifies \$JobGroupID02 runs.

The eighth command uses the Get-SCVirtualHardDisk cmdlet to get the virtual hard disk object named Template.vhd, VHDs on VMHost01 and stores the object in the \$VHD variable.

The ninth command creates a new virtual disk drive and attaches the virtual hard disk stored in \$VHD (Template.vhd) to this new virtual disk drive. The command specifies Bus 0 and LUN 0 on the IDE Bus

---

so that Template.vhd will be attached to the first slot (0) of the Primary Channel (0) on the IDE bus of the new virtual disk drive. The command uses the JobGroup parameter to specify that the new virtual disk drive is not created until just before the New-SCVMTemplate cmdlet runs in the last command.

The tenth command gets the hardware profile object that represents the hardware profile named TempHWProfile from the VMM library and stores the object in the \$HWProfile variable.

The eleventh command gets the specified operating system object (64-bit Edition of Windows Server 2008 R2 Enterprise) and stores the object in the \$OS variable.

The last command creates a virtual machine template named NewTemplate03, sets the owner to Contoso\ReneeLo, specifies that this template will use the hardware profile named TempHWProfile, sets the computer name to be randomly generated (indicated by the asterisk "\*"), and specifies that any virtual machine created by using this template will be joined to the workgroup called WORKGROUP. The New-SCVMTemplate cmdlet uses the JobGroup parameter to specify that all preceding commands that include variable \$JobGroupID02 run before New-SCVMTemplate creates the new template. After Add-SCVirtualHardDisk runs, the resulting virtual hard disk object that is created is automatically associated with the new template.

```
PS C:\> $JobGroupId01 = [Guid]::NewGuid().ToString()
PS C:\> $LogNet = Get-SCLogicalNetwork -Name "LogicalNetwork01"
PS C:\> New-SCVirtualNetworkAdapter -JobGroup $JobGroupID01 -MACAddressType Dynamic -
LogicalNetwork $LogNet
PS C:\> New-SCVirtualSCSIAdapter -JobGroup $JobGroupID01 -AdapterID 6 -Shared $False
PS C:\> New-SCVirtualDVDDrive -JobGroup $JobGroupID01 -Bus 1 -LUN 0
PS C:\> New-SCHardwareProfile -Name "TempHWProfile" -Owner "Contoso\ReneeLo" -Description
"Temporary hardware profile used to create a VM Template" -MemoryMB 512 -JobGroup
$JobGroupID01
PS C:\> $JobGroupId02 = [Guid]::NewGuid().ToString()
PS C:\> $VHD = Get-SCVirtualHardDisk | where { $_.Location -eq
"\VMHost01Share\VHDs\Template.vhd" } | where { $_.HostName -eq "VMHost01.Contoso.com" }
PS C:\> New-SCVirtualDiskDrive -IDE -Bus 0 -LUN 0 -JobGroup $JobGroupID02 -VirtualHardDisk
$VHD
PS C:\> $HWProfile = Get-SCHardwareProfile | where { $_.Name -eq "TempHWProfile" }
PS C:\> $OS = Get-SCOperatingSystem | where { $_.Name -eq "64-bit Edition of Windows Server
2008 R2 Enterprise" }
PS C:\> New-SCVMTemplate -Name "NewTemplate03" -Owner "Contoso\ReneeLo" -HardwareProfile
$HWProfile -JobGroup $JobGroupID02 -ComputerName "*" -JoinWorkgroup "WORKGROUP" -
OperatingSystem $OS -RunAsynchronously
```

#### **4: Create a virtual machine template with the DRProtectionRequired parameter set to \$True.**

The first command gets the operating system named Windows Server 2012 and stores it in the \$OS variable.

The second command gets the virtual hard disk object named VHDX01 from the VMM library and stores the object in the \$VHDX variable.

---

The last command creates a virtual machine template named VMTemplate04 that requires DR protection for the virtual machines created from this template.

```
PS C:\> $OS = Get-SCOperatingSystem -VMMServer "VMMServer01.Contoso.com" | where {$_.Name -eq "Windows Server 2012"}
PS C:\> $VHDX = Get-SCVirtualHardDisk -VMMServer "VMMServer01.Contoso.com" | where { $_.Name -eq "VHDX01.vhdx" -and $_.LibraryServer.Name -eq "LibServer01.Contoso.com" }
PS C:\> New-SCVMTemplate -Name "VMTemplate04" -VirtualHardDisk $VHDX -OperatingSystem $OS -DRProtectionRequired $True
```

## 5: Create a customizable Linux virtual machine template from a virtual hard disk.

The first command gets the specified operating system object (CentOS Linux 6 (64 bit)) and stores the object in the \$OS variable.

The second command gets the virtual hard disk object named LinuxVHD01 from the VMM library on LibServer01 and stores the object in the \$VHD variable.

The last command creates a virtual machine template named LinuxVMTemplate01 from LinuxVHD01 and specifies the name of the operating system.

```
PS C:\> $OS = Get-SCOperatingSystem -VMMServer "VMMServer01.Contoso.com" | where {$_.Name -eq "CentOS Linux 6 (64 bit)"}
PS C:\> $VHD = Get-SCVirtualHardDisk -VMMServer "VMMServer01.Contoso.com" | where { $_.Name -eq "LinuxVHD01.vhd" -and $_.LibraryServer.Name -eq "LibServer01.Contoso.com" }
PS C:\> New-SCVMTemplate -Name "LinuxVMTemplate01" -VirtualHardDisk $VHD -OperatingSystem $OS
```

## Related topics

[Get-SCVMTemplate](#)

[New-SCP2V](#)

[Remove-SCVMTemplate](#)

[Set-SCVMTemplate](#)



---

# New-SCVMXComputerConfiguration

---

## New-SCVMXComputerConfiguration

Creates a VMX computer configuration object by gathering virtual machine configuration information from a virtual machine created in VMware that you plan to convert to a virtual machine deployed on a Windows-based host managed by VMM.

### Syntax

Parameter Set: Default

```
New-SCVMXComputerConfiguration [-VMXPath] <String> [-JobVariable <String> ] [-LibraryServer <LibraryServer> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The New-SCVMXComputerConfiguration cmdlet creates a VMX computer configuration object by gathering information about the physical characteristics of a VMware-based virtual machine and its disks that you plan to convert to a virtual machine deployed on a Windows-based Hyper-V host managed by Virtual Machine Manager (VMM). This cmdlet does not collect information about the operating system or data on the VMware-based virtual machine.

VMWare virtual hard disk formats supported by the New-SCVMXComputerConfiguration cmdlet include:

- monolithicSparse
- monolithicFlat
- vmfs
- twoGbMaxExtentSparse
- twoGbMaxExtentFlat

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | false |

|                             |       |
|-----------------------------|-------|
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-LibraryServer<LibraryServer>**

Specifies a VMM library server object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | false |

|                             |       |
|-----------------------------|-------|
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-VMXPath<String>**

Specifies the full UNC path to the .vmx file of a VMware virtual machine.

Example format: \\ServerName\VolumeName\DirectoryName\VMwareVM.vmx

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | 1     |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

---

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **VMXMachineConfig**

## Examples

### 1: Gather information from a VMware-based virtual machine.

The first command gets the library server object named LibServer01 in the Contoso.com domain and stores the object in the \$LibServ variable.

The last command gathers the machine configuration information for the .vmx file located at "\\FileServer01\MSSCVMMLibrary\VMware\VMSource.vmx" on the library server. The New-SCVMXComputerConfiguration cmdlet stores the resulting VMX computer configuration object associated with VMSource.vmx in the VMM database.

NOTE: If you look in Library view in the VMM console, you cannot see the file VMSource.vmx file because the .vmx file is part of a single virtual machine object. What you see in Library view is the virtual machine. To find the path to a .vmdk file, view the properties for that virtual machine.

```
PS C:\> $LibServ = Get-SCLibraryServer -ComputerName "LibServer01.Contoso.com"
```

```
PS C:\> New-SCVMXComputerConfiguration -LibraryServer $LibServ -VMXPath  
"\\FileServer01\MSSCVMMLibrary\VMware\VMSource.vmx"
```

## Related topics

[Add-SCPatch](#)

[Copy-SCVirtualHardDisk](#)

[Get-SCVMXComputerConfiguration](#)

[New-SCV2V](#)

[Remove-SCVMXComputerConfiguration](#)

---

# Publish-SCWindowsPE

---

## Publish-SCWindowsPE

Publishes an updated Windows PE image for use by all PXE servers in your VMM environment.

### Syntax

Parameter Set: UseCustomizedWinPE

```
Publish-SCWindowsPE -Path <String> [-ISOPath <String> ] [-IsUEFI] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: UseDefaultImage

```
Publish-SCWindowsPE -UseDefaultImage [-ISOPath <String> ] [-IsUEFI] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Publish-SCWindowsPE cmdlet publishes an updated Windows Preinstallation Environment (Windows PE) image for use by all Pre-Boot Execution Environment (PXE) servers in your Virtual Machine Manager (VMM) environment.

#### SCENARIOS THAT REQUIRE AN UPDATED WINDOWS PE IMAGE

- The Windows Automated Installation Kit (AIK) is patched, or Microsoft issues a new version of Windows AIK.
- The VMM agent binaries are patched.
- You add your own drivers, or other custom files, to Windows PE.

NOTE: Customize the Windows PE image by adding drivers or custom files using standard Windows tools and then use this cmdlet to publish the updated Windows PE boot WIM image on a library share.

Each scenario requires that you use this cmdlet not only to add the updated Windows PE image to VMM but also to rebuild it.

#### TASKS YOU PERFORM WITH THIS CMDLET

1. Specify the source Windows PE image:

Option 1: Start from the standard Windows PE image from the Windows AIK that is currently installed on the VMM management server. In this case, both x86 and x64 versions of Windows PE must be processed.

Option 2: Start from an existing Windows PE image on a VMM library share.

---

In this case, only the specified Windows PE image (which is either x86 or x64) is processed.

NOTE: The Windows PE image chosen must not already contain the VMM agent. If such a Windows PE image is chosen, an error is returned and the Windows PE image is not imported.

2. Specify the Library Resource Folder in which to store the updated Windows PE image.

NOTE: VMM creates a folder named "Boot WIMS with Agent" on the target Library Resource Folder (if it does not already exist).

3. Construct a new Windows PE image from the source Windows PE image as follows:

a. Copy the source Windows PE image into a temporary location on the VMM management server.

b. Mount the WinPE image.

c. Copy the agent files from the fixed location on the VMM management server into a fixed location in the Windows PE image. Overwrite any files that already exist in the Windows PE image, but do not otherwise delete any files or directories.

d. Perform Windows PE configuration tasks, such as setting the RAM disk size, ensuring that optional features like WMI are installed, and so on.

e. Unmount the image and commit changes.

f. Copy the updated Windows PE image into the "Boot WIMs with Agent" folder.

4. Force discovery on the Library Resource Folder, and confirm that the newly placed Windows PE image appears in your VMM environment.

5. Use Publish-SCWindowsPE to copy all Windows PE images in the "Boot WIMs with Agent" folder to all PXE servers, and to extract Windows network boot programs (NBP) on each PXE server.

## Parameters

### **-ISOPath<String>**

Specifies the destination path for an ISO file.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-IsUEFI**

Indicates that the computer on which the operating system will be installed is Unified Extensible Firmware Interface (UEFI)-based.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-Path<String>**

Specifies the destination path for the operation.

Example formats:

Local path     -Path "F:\"

UNC path       -Path "\\Library\Templates"

Volume GUID path -Path "\\?\Volume{4703c1ea-8ae7-11db-b473-00123f7603e3}\"

VMware ESX path -Path "[storage1]\MyVMwareFolderForVMs\MyVM.vmx"

Citrix XenServer path - Path "Local storage[99b6212f-b63d-c676-25f9-d6c460992de7]"

Wildcards are supported for "Get" cmdlets and when you specify the UNC path:

Example format:

UNC path       -Path "\\VMHostServer\MyVMs\\*VM\*"

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-UseDefaultImage**

Indicates that new or updated Windows Preinstallation Environment (Windows PE) images are published by using the standard Windows PE images from the latest Windows Automated Deployment Kit (Windows ADK).

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |



---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Examples

### 1: Update the Winows PE image with a custom Windows PE image base.

This command uses a customized base image to create a Windows Preinstallation Environment (Windows PE) image and updates all VMM Pre-Boot Execution Environment (PXE) servers.

```
PS C:\> Publish-SCWindowsPE -Path "\\LibraryServer02\VMMWinPE\ContosoIT.wim"
```

### 2: Re-create the Winows PE image and update the VMM PXE servers.

This command re-creates the Windows PE image by using the Windows PE image from (or updated by) the Windows ADK. It then updates all VMM PXE servers.

```
PS C:\> Publish-SCWindowsPE -UseDefaultImage
```

---

# Read-SCGuestInfo

---

## Read-SCGuestInfo

Retrieves the value associated with a key in a guest operating system.

### Syntax

Parameter Set: MultipleKvpKeys

```
Read-SCGuestInfo [-VM] <VM> -KvpMap <Hashtable> [ <CommonParameters>]
```

Parameter Set: SingleKvpKey

```
Read-SCGuestInfo [-VM] <VM> [-Key] <String> [ <CommonParameters>]
```

### Detailed Description

The Read-SCGuestInfo cmdlet retrieves the value associated with a key (key/value pair) in a guest operating system.

### Parameters

#### **-Key<String>**

Specifies the key in a key/value pair (KVP).

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | 2     |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### **-KvpMap<Hashtable>**

Specifies a hashtable of key/value pairs (KVPs) corresponding to the KVP values exposed by Hyper-V.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VM<VM>**

Specifies a virtual machine object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **String**

## **Examples**

### **1: Get the IntegrationServicesVersion value for a specified key for a virtual machine.**

The first command gets the virtual machine object named \$VM01 and stores the object in the \$VM variable.

The second command returns the IntegrationServicesVersion key/value pair for virtual machine VM01.

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM01"
```

---

```
PS C:\> Read-SCGuestInfo -VM $VM -Key "FullyQualifiedDomainName"
```

## 2: Get the IntegrationServicesVersion value for a specified key for a virtual machine by using the pipeline operator.

This command returns the IntegrationServicesVersion key/value pair for virtual machine VM01.

```
PS C:\> Get-SCVirtualMachine -Name "VM01" | Read-SCGuestInfo -Key IntegrationServicesVersion
```

## 3: Get multiple KVP values based on specified keys for a virtual machine.

The first command creates an array named \$ValuesMap.

The second and third commands add values to the \$ValuesMap array.

The fourth command gets the virtual machine object named VM02 and stores the object in the \$VM variable.

The last command returns the IntegrationServicesVersion and NetworkAddressIPv4 key/value pairs for virtual machine VM02.

```
PS C:\> $ValuesMap = @{}
PS C:\> $ValuesMap.Add("NetworkAddressIPv4", $null)
PS C:\> $ValuesMap.Add("IntegrationServicesVersion", $null)
PS C:\> $VM = Get-SCVirtualMachine "VM02"
PS C:\> Read-SCGuestInfo -VM $vm -KvpMap $ValuesMap
```

## 4: Read multiple data types through a hashtable.

The first command gets the virtual machine object named VM03 and stores the object in the \$VM variable.

The second command creates an array named \$ValuesMap.

The third command adds NetworkAddressIPv4 to the \$ValuesMap array.

The fourth command gets the NetworkAddressIPv4 key/value pair for VM03.

The fifth command adds NetworkAddressIPv6 to the \$ValuesMap array.

The sixth command gets the NetworkAddressIPv6 key/value pair for VM03.

The seventh command creates an array named \$ValuesMap2 which contains NetworkAddressIPv4 and FullyQualifiedDomainName.

The last command returns the key/value pairs for NetworkAddressIPv4 and FullyQualifiedDomainName for VM03.

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM03"
PS C:\> $ValuesMap = @{}
PS C:\> $ValuesMap.Add("NetworkAddressIPv4", $null)
```

---

```
PS C:\> Read-SCGuestInfo -VM $VM -KvpMap $ValuesMap
PS C:\> $ValuesMap.Add("NetworkAddressIPv6", $null)
PS C:\> Read-SCGuestInfo -VM $VM -KvpMap $ValuesMap
PS C:\> $ValuesMap2 = @{"NetworkAddressIPv4" = $null; "FullyQualifiedDomainName" = $null}
PS C:\> Read-SCGuestInfo -VM $VM -KvpMap $ValuesMap2
```

## 5: Read keys that do not exist.

The first command gets the virtual machine object named VM01 and stores the object in the \$VM variable.

The second, fourth, and sixth commands each create a set of keys that are null and stores the set in the \$KeysDoNotExist variable.

The third, fifth, and seventh commands read the KVPMAP in \$KeysDoNotExist and displays the results to the user.

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM01"
PS C:\> $KeysDoNotExist = @{"o1ff1" = $null; "o1ff2" = $null; "o1ff3" = $null ; "o1ff4" = $null }
PS C:\> Read-SCGuestInfo -VM $VM -KVPMAP $KeysDoNotExist
PS C:\> $KeysDoNotExist = @{"off4" = $null; "o1ff2" = $null; "o1ff3" = $null ; "o1ff4" = $null }
PS C:\> Read-SCGuestInfo -VM $VM -KVPMAP $KeysDoNotExist
PS C:\> $KeysDoNotExist = @{"o1ff1" = $null; "o1ff2" = $null; "off4" = $null ; "o1ff4" = $null }
PS C:\> Read-SCGuestInfo -VM $VM -KVPMAP $KeysDoNotExist
```

## Related topics

[Set-SCGuestInfo](#)

---

# Read-SCLibraryShare

---

## Read-SCLibraryShare

Updates the state and metadata of VMM library objects stored in a library share.

### Syntax

Parameter Set: Default

```
Read-SCLibraryShare [-LibraryShare] <LibraryShare> [-JobVariable <String> ] [-Path <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

### Detailed Description

The Read-SCLibraryShare cmdlet updates the state and metadata of all VMM library objects stored in the specified library share. This update also finds new library files on the specified library share as well as new child shared folders under the specified library share.

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### **-LibraryShare<LibraryShare>**

Specifies a VMM library share object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-Path<String>**

Specifies the destination path for the operation.

Example formats:

Local path -Path "F:\"

UNC path -Path "\\Library\Templates"

Volume GUID path -Path "\\?\Volume{4703c1ea-8ae7-11db-b473-00123f7603e3}\\"

VMware ESX path -Path "[storage1]MyVMwareFolderForVMs\MyVM.vmx"

Citrix XenServer path - Path "Local storage[99b6212f-b63d-c676-25f9-d6c460992de7]"

Wildcards are supported for "Get" cmdlets and when you specify the UNC path:

Example format:

UNC path -Path "\\VMHostServer\MyVMs\\*VM\*"

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | false |
| Position? | named |

|                             |       |
|-----------------------------|-------|
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **LibraryShare**

## **Examples**

### **1: Update a specified library share.**

The first command gets the library share object named AllVHDs on LibraryServer01 from the VMM library on VMMServer01 and then stores the object in the \$LibShare variable.

The second command updates the state and metadata information for all library objects in the share stored in \$LibShare, and then it adds any new library objects found in the share to the VMM library.

```
PS C:\> $LibShare = Get-SCLibraryShare -VMMServer "VMMServer01.Contoso.com" | where {
$_.LibraryServer.Name -eq "LibraryServer01.Contoso.com" -and $_.Name -eq "AllVHDs" }
PS C:\> Read-SCLibraryShare -LibraryShare $LibShare
```



---

## 2: Update multiple library shares.

The first command gets the library share objects on LibraryServer01 with the string "vhd" in their names, and then stores the objects in the \$LibShares variable.

The second command updates the information for all library shares stored in \$LibShares, and then it adds any new library objects found in these shares to the VMM library.

```
PS C:\> $LibShares = Get-SCLibraryShare -VMMServer "VMMServer1.Contoso.com" | where {
$_.LibraryServer.Name -eq "LibraryServer01.Contoso.com" -and $_.Name -match "vhd" }
PS C:\> $LibShares | Read-SCLibraryShare
```

## 3. Update a specific subdirectory on a library share.

The first command gets the library share object named AllVHDs on LibraryServer01 from the VMM library on VMMServer01 and then stores the object in the \$LibShare variable.

The second command updates the state and metadata information for all library objects in the specified subdirectory on the library share that is stored in \$LibShare, and then it adds any new library objects found in the share to the VMM library.

```
PS C:\> $LibShare = Get-SCLibraryShare -VMMServer "VMMServer01.Contoso.com" | where {
$_.LibraryServer.Name -eq "LibraryServer01.Contoso.com" -and $_.Name -eq "AllVHDs" }
PS C:\> Read-SCLibraryShare -LibraryShare $LibShare -Path "Production\WebFrontEnd"
```

## Related topics

[Add-SCLibraryShare](#)

[Find-SCLibraryShare](#)

[Get-SCLibraryShare](#)

[Remove-SCLibraryShare](#)

[Set-SCLibraryShare](#)

---

# Read-SCLoadBalancer

---

## Read-SCLoadBalancer

Refreshes load balancer information in the VMM console.

### Syntax

Parameter Set: Default

```
Read-SCLoadBalancer [-LoadBalancer] <LoadBalancer> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Read-SCLoadBalancer cmdlet refreshes load balancer information and reflects the updates in the Virtual Machine Manager (VMM) console.

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### **-LoadBalancer<LoadBalancer>**

Specifies a load balancer object.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

|                             |                |
|-----------------------------|----------------|
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

---

|                             |                |
|-----------------------------|----------------|
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **LoadBalancer**

## Examples

### 1: Refresh a load balancer.

The first command gets the load balancer object stored at LB01.Contoso.com and stores the object in the \$LoadBalancer variable.

The second command refreshes the load balancer stored in \$LoadBalancer.

```
PS C:\> $LoadBalancer = Get-SCLoadBalancer -LoadBalancerAddress "LB01.Contoso.com"
```

```
PS C:\> Read-SCLoadBalancer -LoadBalancer $LoadBalancer
```

## Related topics

[Add-SCLoadBalancer](#)

[Get-SCLoadBalancer](#)

[Remove-SCLoadBalancer](#)

[Set-SCLoadBalancer](#)

[Test-SCLoadBalancer](#)

---

# Read-SCLoadBalancerVIP

---

## Read-SCLoadBalancerVIP

Refreshes load balancer VIP information in the VMM console.

### Syntax

Parameter Set: Default

```
Read-SCLoadBalancerVIP [-LoadBalancerVIP] <LoadBalancerVIP> [-JobVariable <String> ] [-  
PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [  
<CommonParameters>]
```

### Detailed Description

The Read-SCLoadBalancerVIP cmdlet refreshes load balancer virtual IP (VIP) information and reflects the updates in the Virtual Machine Manager (VMM) console.

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### **-LoadBalancerVIP<LoadBalancerVIP>**

Specifies a virtual IP (VIP) in a load balancer.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

---

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **LoadBalancerVIP**

## Examples

### 1: Refresh a specific load balancer virtual IP (IP).

The first command gets the load balancer object with the address LB01.Contoso.com and stores the object in the \$LoadBalancer variable.

The second command gets the load balancer VIP for the load balancer stored in \$LoadBalancer with the IP address of 10.0.0.1.

The last command refreshes the load balancer VIP stored in \$VIP.

```
PS C:\> $LoadBalancer = Get-SCLoadBalancer -LoadBalancerAddress "LB01.Contoso.com"
PS C:\> $VIP = Get-SCLoadBalancerVIP -LoadBalancer $LoadBalancer -IPAddress "10.0.0.1"
PS C:\> Read-SCLoadBalancerVIP -LoadBalancerVIP $VIP
```

## Related topics

[Get-SCLoadBalancerVIP](#)

[New-SCLoadBalancerVIP](#)

[Remove-SCLoadBalancerVIP](#)

---

# Read-SCNetworkGateway

---

## Read-SCNetworkGateway

Refreshes a network gateway object.

### Syntax

Parameter Set: Default

```
Read-SCNetworkGateway [-NetworkGateway] <NetworkGateway> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Read-SCNetworkGateway cmdlet refreshes a network gateway object.

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### **-NetworkGateway<NetworkGateway>**

Specifies a network gateway object.

To obtain a network gateway object, use the Get-SCNetworkGateway cmdlet.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|



|                             |                |
|-----------------------------|----------------|
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

---

|                             |                |
|-----------------------------|----------------|
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Related topics

[Add-SCNetworkGateway](#)

[Get-SCNetworkGateway](#)

[Remove-SCNetworkGateway](#)

[Set-SCNetworkGateway](#)

[Test-SCNetworkGateway](#)

---

# Read-SCService

---

## Read-SCService

Refreshes the information for a service.

### Syntax

Parameter Set: Default

```
Read-SCService [-Service] <Service> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Read-SCService cmdlet refreshes the information for a service and reflects the updates in the Virtual Machine Manager (VMM) console.

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Service<Service>**

Specifies a VMM service object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

---

|                             |                |
|-----------------------------|----------------|
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Examples

### 1: Refresh a service object.

The first command gets the service object named Contoso Service Configuration 01 and stores the object in the \$Service variable.

The second command refreshes the properties of the service stored in \$Service.

```
PS C:\> $Service = Get-SCService -Name "Service01"
```

```
PS C:\> Read-SCService -Service $Service
```

## Related topics

[Get-SCService](#)

[New-SCService](#)

[Remove-SCService](#)

[Resume-SCService](#)

[Set-SCService](#)

[Start-SCService](#)

[Stop-SCService](#)

[Suspend-SCService](#)

[Update-SCService](#)

---

# Read-SCServiceTemplate

---

## Read-SCServiceTemplate

Refreshes the properties of a service template and displays the updates in the VMM console.

### Syntax

Parameter Set: Default

```
Read-SCServiceTemplate [-ServiceTemplate] <ServiceTemplate> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

### Detailed Description

The Read-SCServiceTemplate cmdlet refreshes the properties of a service template and reflects the updates in the Virtual Machine Manager (VMM) console.

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-ServiceTemplate<ServiceTemplate>**

Specifies a service template object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

---

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **ServiceTemplate**

## Examples

### 1: Refresh a specified service template object.

The first command gets the service template object named ServiceTemplate01 and stores the object in the \$SvcTemplate variable.

The second command refreshes the properties of the service template stored in \$SvcTemplate.

```
PS C:\> $SvcTemplate = Get-SCServiceTemplate -VMMServer "VMMServer01.Contoso.com" -Name "ServiceTemplate01"
```

```
PS C:\> Read-SCServiceTemplate -ServiceTemplate $SvcTemplate
```

## Related topics

[Get-SCServiceTemplate](#)

[New-SCServiceTemplate](#)

[Remove-SCServiceTemplate](#)

[Resolve-SCServiceTemplate](#)

[Set-SCServiceTemplate](#)

[Test-SCServiceTemplate](#)



---

# Read-SCStorageProvider

---

## Read-SCStorageProvider

Retrieves updated information from the storage provider including array, pool, and logical unit information exposed by the provider.

### Syntax

Parameter Set: Default

```
Read-SCStorageProvider [-StorageProvider] <StorageProvider> [-Force] [-JobVariable <String>]  
] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

### Detailed Description

The Read-SCStorageProvider cmdlet retrieves updated information from the storage provider including array, pool, and logical unit information exposed by the provider.

### Parameters

#### -Force

Forces the operation to complete.

For example:

- Remove-SCSCVMHost -Force

Forces the removal of a host object from the VMM database.

- Stop-SCVirtualMachine -Force

Stops a virtual machine.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-StorageProvider<StorageProvider>**

Specifies a storage provider object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

### **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **StorageProvider**

### **Examples**

#### **1: Refresh information about a storage provider.**

The first command gets the storage provider named StorProv01 and stores it in the \$Provider variable. The second command retrieves updated information about the storage provider stored in the \$Provider variable.

```
PS C:\> $Provider = Get-SCStorageProvider -Name "StorProv01.Contoso.com"
```

```
PS C:\> Read-SCStorageProvider -StorageProvider $Provider
```

### **Related topics**

[Add-SCStorageProvider](#)

[Get-SCStorageProvider](#)

[Remove-SCStorageProvider](#)

[Set-SCStorageProvider](#)

---

# Read-SCVirtualizationManager

---

## Read-SCVirtualizationManager

Refreshes the properties of a VMware vCenter Server so that the VMM console displays updated information about vCenter Server entities.

### Syntax

Parameter Set: Default

```
Read-SCVirtualizationManager [-VirtualizationManager] <VirtualizationManager> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

### Detailed Description

The Read-SCVirtualizationManager cmdlet refreshes the properties of a VMware vCenter Server managed by Virtual Machine Manager (VMM) so that the VMM console displays updated information about vCenter Server entities.

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VirtualizationManager<VirtualizationManager>**

Specifies a virtualization manager object managed by VMM.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

---

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **VirtualizationManager**

## Notes

- Requires a VMM virtualization manager object, which can be retrieved by using the Get-SCVirtualizationManager cmdlet.

## Examples

### 1: Refresh all VMware vCenter Servers managed by VMM.

The first command retrieves all virtualization manager objects from the VMM database on VMMServer01 and stores the returned objects in the \$VirtManagers array.

The second command uses a foreach loop statement to refresh the properties of the objects stored in \$VirtManagers so that current information about these virtualization managers displays in the VMM console.

For more information about the standard Windows PowerShell foreach loop statement, type: Get-Help about\_ForEach.

```
PS C:\> $VirtManagers = Get-SCVirtualizationManager -VMMServer "VMMServer01.Contoso.com"
PS C:\> ForEach ($VManager in $VirtManagers) {Read-SCVirtualizationManager -
VirtualizationManager $VManager}
```

### 2: Refresh a specific VMware vCenter Server managed by VMM.

The first command gets the virtualization manager object named VirtMgrServer01 from VMMServer01 and stores the object in \$VirtManager.

The second command refreshes the properties for the object stored in \$VirtManager so that current information about this virtualization manager displays in the VMM console.

```
PS C:\> $VirtManager = Get-SCVirtualizationManager -VMMServer "VMMServer01.Contoso.com" -
Computername "VirtMgrServer01.Contoso.com"
PS C:\> Read-SCVirtualizationManager -VirtualizationManager $VirtManager
```

## Related topics

[Get-SCVirtualizationManager](#)

[Add-SCVirtualizationManager](#)

[Remove-SCVirtualizationManager](#)

[Set-SCVirtualizationManager](#)

---

# Read-SCVirtualMachine

---

## Read-SCVirtualMachine

Refreshes the properties of a virtual machine so that the VMM console displays updated information about the virtual machine.

### Syntax

Parameter Set: FullRefreshSingleVm

```
Read-SCVirtualMachine [-VM] <VM> [-Force] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

Parameter Set: LightRefreshVmsOnHost

```
Read-SCVirtualMachine [-VMHost] <Host> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

### Detailed Description

The Read-SCVirtualMachine cmdlet refreshes the properties of a virtual machine so that the Virtual Machine Manager (VMM) console displays updated information about the virtual machine. The updated properties include Name, Status, Host, Owner, CPUAverage, Service, OperatingSystem, and other properties.

### Parameters

#### -Force

Forces the operation to complete.

For example:

- Remove-SCSCVMHost -Force

Forces the removal of a host object from the VMM database.

- Stop-SCVirtualMachine -Force

Stops a virtual machine.

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | false |
| Position? | named |

---

|                             |       |
|-----------------------------|-------|
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | false |
| Position? | named |



|                             |       |
|-----------------------------|-------|
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VM<VM>**

Specifies a virtual machine object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-VMHost<Host>**

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

### **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **VirtualMachine**

---

## Notes

- Requires a virtual machine object, which can be retrieved by using the Get-SCVirtualMachine cmdlet.

## Examples

### 1: Refresh information about a specific virtual machine.

The first command gets the virtual machine object named VM01 and stores the object in the \$VM variable.

The second command refreshes the properties of the virtual machine stored in \$VM (in this case, VM01). After this command completes successfully, current information about this virtual machine will appear in the VMM console.

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM01"  
PS C:\> Read-SCVirtualMachine -VM $VM
```

### 2: Refresh all virtual machines on hosts whose name matches the specified string.

The first command gets all virtual machine objects from VMMServer01 deployed on hosts whose name contains the string "VMM", and then stores the virtual machine objects in the \$VMs variable.

The second command refreshes the properties of each virtual machine object stored in \$VMs.

```
PS C:\> $VMs = Get-SCVirtualMachine -VMMServer "VMMServer01.Contoso.com" | where {  
$_.VMHost.Name -match "VMM" }  
PS C:\> $VMs | Read-SCVirtualMachine
```

## Related topics

[Get-SCVirtualMachine](#)

[Move-SCVirtualMachine](#)

[New-SCVirtualMachine](#)

[Register-SCVirtualMachine](#)

[Remove-SCVirtualMachine](#)

[Repair-SCVirtualMachine](#)

[Reset-SCVirtualMachine](#)

[Resume-SCVirtualMachine](#)

[Set-SCVirtualMachine](#)

[Start-SCVirtualMachine](#)

[Stop-SCVirtualMachine](#)

[Suspend-SCVirtualMachine](#)

---

# Read-SCVirtualSwitchExtensionManager

---

## Read-SCVirtualSwitchExtensionManager

Refreshes a virtual switch extension manager.

### Syntax

Parameter Set: Default

```
Read-SCVirtualSwitchExtensionManager [-VirtualSwitchExtensionManager]
<VirtualSwitchExtensionManager> [-JobVariable <String> ] [-PROTipID <Guid> ] [-
RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Read-SCVirtualSwitchExtensionManager cmdlet refreshes a virtual switch extension manager.

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VirtualSwitchExtensionManager<VirtualSwitchExtensionManager>**

Specifies a virtual switch extension manager object.

To obtain a virtual switch extension manager object, use the Get-SCVirtualSwitchExtensionManager cmdlet.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

---

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

### **Related topics**

[Add-SCVirtualSwitchExtensionManager](#)

[Get-SCVirtualSwitchExtensionManager](#)

[Remove-SCVirtualSwitchExtensionManager](#)

[Set-SCVirtualSwitchExtensionManager](#)

[Test-SCVirtualSwitchExtensionManager](#)

---

# Read-SCVMHost

---

## Read-SCVMHost

Refreshes virtual machine host properties in the VMM Console.

### Syntax

Parameter Set: DefaultRefresh

```
Read-SCVMHost [-VMHost] <Host> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

Parameter Set: OutOfBandRefresh

```
Read-SCVMHost [-VMHost] <Host> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RefreshOutOfBandProperties] [-RunAsynchronously] [ <CommonParameters>]
```

### Detailed Description

The Read-SCVMHost cmdlet refreshes the properties of a virtual machine host so that the VMM Console displays updated information about the host.

Host properties that this cmdlet updates include:

- Name
- Operating system
- Status (such as Responding)
- Host volumes (typically, addition or removal of drive letters, mount points, as well as used and available space)
- Network adapters (addition or removal)

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | false |
| Position? | named |

---

|                             |       |
|-----------------------------|-------|
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RefreshOutOfBandProperties**

Refreshes available computer hardware properties through the out-of-band baseboard management controller (BMC). This process is different from the regular host refreshing logic, because the host refresher goes through the VMM agent inside the operating system (in-band). The properties that are refreshed through BMC are different from those in the in-band refresher.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VMHost<Host>**

Specifies a virtual machine host object. VMM supports Hyper-V hosts, VMware ESX hosts, and Citrix XenServer hosts.

For more information about each type of host, type: "Get-Help Add-SCVMHost -detailed". See the examples for a specific cmdlet to determine how that cmdlet uses this parameter.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **Host**

## **Examples**

### **1: Refresh information about a specific host.**

The first command gets the host object named VMHost01 and stores the object in the \$VMHost variable.



---

The second command refreshes the properties for VMHost01 so that current information about this host will appear in the VMM Console.

```
PS C:\> $VMHost = Get-SCVMHost -ComputerName "VMHost01"  
PS C:\> Read-SCVMHost -VMHost $VMHost
```

## 2: Refresh information about all hosts.

This command refreshes information about all hosts managed by VMMServer01 so that current information about each host will appear in the VMM Console.

```
PS C:\> Get-SCVMHost -VMMServer "VMMServer01.Contoso.com" | Read-SCVMHost
```

## 3: Refresh information about a given host through its OOB channel.

This command refreshes information about host VMHost02 using its out-of-band interface so that current information about the host appears in the VMM console.

```
PS C:\> Get-SCVMHost -ComputerName "VMHost02" | Read-SCVMHost -RefreshOutOfBandProperties
```

## Related topics

[Add-SCVMHost](#)

[Get-SCVMHost](#)

[Read-SCLibraryShare](#)

[Read-SCVirtualMachine](#)

[Remove-SCVMHost](#)

[Set-SCVMHost](#)

---

# Read-SCVMHostCluster

---

## Read-SCVMHostCluster

Refreshes host cluster properties in the VMM console.

### Syntax

Parameter Set: Default

```
Read-SCVMHostCluster [-VMHostCluster] <HostCluster> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

### Detailed Description

The Read-SCVMHostCluster cmdlet refreshes host cluster properties so that the Virtual Machine Manager (VMM) console displays updated information about the host cluster.

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

---

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMHostCluster<HostCluster>**

Specifies a VMM host cluster object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

---

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **VMHostCluster**

## Examples

### 1: Refresh information about a specific host cluster.

This command gets the host cluster object named VMHostCluster01 and passes the object to the Refresh-SCVMHostCluster cmdlet which refreshes the properties for the host cluster so that current information about the host cluster appears in the VMM console.

```
PS C:\> Get-SCVMHostCluster -Name "VMHostCluster01.Contoso.Com" | Read-SCVMHostCluster
```

## Related topics

[Add-SCVMHostCluster](#)

[Find-SCCluster](#)

[Get-SCVMHostCluster](#)

[Install-SCVMHostCluster](#)

[Move-SCVMHostCluster](#)

[Remove-SCVMHostCluster](#)

[Set-SCVMHostCluster](#)

[Test-SCVMHostCluster](#)

[Uninstall-SCVMHostCluster](#)

---

# Read-SCVMNetworkGateway

---

## Read-SCVMNetworkGateway

Refreshes a VM network gateway.

### Syntax

Parameter Set: Default

```
Read-SCVMNetworkGateway [-VMNetworkGateway] <VMNetworkGateway> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Read-SCVMNetworkGateway cmdlet refreshes a VM network gateway.

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

---

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-VMNetworkGateway<VMNetworkGateway>**

Specifies a VM network gateway object.

To obtain a VM network gateway object, use the Get-SCVMNetworkGateway cmdlet.

---

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

### Related topics

[Add-SCVMNetworkGateway](#)

[Get-SCVMNetworkGateway](#)

[Remove-SCVMNetworkGateway](#)

[Set-SCVMNetworkGateway](#)

---

# Register-SCStorageFileShare

---

## Register-SCStorageFileShare

Registers a storage file share with a VM host, cluster, or library server.

### Syntax

Parameter Set: RegisterToHost

```
Register-SCStorageFileShare -FileSharePath <String> -VMHost <Host> [-JobGroup <Guid> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

Parameter Set: RegisterShareToCluster

```
Register-SCStorageFileShare -StorageFileShare <StorageFileShare> -VMHostCluster <HostCluster> [-JobGroup <Guid> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

Parameter Set: RegisterShareToHost

```
Register-SCStorageFileShare -StorageFileShare <StorageFileShare> -VMHost <Host> [-JobGroup <Guid> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

Parameter Set: RegisterShareToLibrary

```
Register-SCStorageFileShare -LibraryServer <LibraryServer> -StorageFileShare <StorageFileShare> [-JobGroup <Guid> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

Parameter Set: RegisterToCluster

```
Register-SCStorageFileShare -FileSharePath <String> -VMHostCluster <HostCluster> [-JobGroup <Guid> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

Parameter Set: RegisterToLibrary

```
Register-SCStorageFileShare -FileSharePath <String> -LibraryServer <LibraryServer> [-JobGroup <Guid> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

### Detailed Description

The Register-SCStorageFileShare cmdlet registers a storage file share with a virtual machine host, cluster, or library server.



---

## Parameters

### **-FileSharePath<String>**

Specifies the path to the file share.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobGroup<Guid>**

Specifies an identifier for a series of commands that will run as a set just before the final command that includes the same job group identifier runs.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                        |       |
|------------------------|-------|
| Aliases                | none  |
| Required?              | false |
| Position?              | named |
| Default Value          | none  |
| Accept Pipeline Input? | false |

---

|                             |       |
|-----------------------------|-------|
| Accept Wildcard Characters? | false |
|-----------------------------|-------|

### **-LibraryServer<LibraryServer>**

Specifies a VMM library server object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                        |       |
|------------------------|-------|
| Aliases                | none  |
| Required?              | false |
| Position?              | named |
| Default Value          | none  |
| Accept Pipeline Input? | false |

---

|                             |       |
|-----------------------------|-------|
| Accept Wildcard Characters? | false |
|-----------------------------|-------|

### **-StorageFileShare<StorageFileShare>**

Specifies a storage file share.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMHost<Host>**

Specifies a virtual machine host object. VMM supports Hyper-V hosts, VMware ESX hosts, and Citrix XenServer hosts.

For more information about each type of host, type: "Get-Help Add-SCVMHost -detailed". See the examples for a specific cmdlet to determine how that cmdlet uses this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMHostCluster<HostCluster>**

Specifies a VMM host cluster object.

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | true  |
| Position? | named |

---

|                             |       |
|-----------------------------|-------|
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **StorageFileShare**

## Examples

### 1: Register a storage file share with a library server.

The first command gets the storage file share object named FileShare01 and stores the object in the \$FileShare variable.

The second command gets the VMM library server object named LibraryServer01 and stores the object in the \$LibServer variable.

The last command registers FileShare01 with LibraryServer01.

```
PS C:\> $FileShare = Get-SCStorageFileShare -Name "FileShare01"
```

```
PS C:\> $LibServer = Get-SCLibraryServer -Computername "LibraryServer01.Contoso.com"
```

```
PS C:\> Register-SCStorageFileShare -StorageFileShare $FileShare -LibraryServer $LibServer
```

## Related topics

[Get-SCStorageFileShare](#)

[New-SCStorageFileShare](#)

[Remove-SCStorageFileShare](#)

[Repair-SCStorageFileShare](#)

[Set-SCStorageFileShare](#)

[Unregister-SCStorageFileShare](#)

---

# Register-SCStorageLogicalUnit

---

## Register-SCStorageLogicalUnit

Associates a logical unit with a host.

### Syntax

Parameter Set: RegisterToClusterJobGroup

```
Register-SCStorageLogicalUnit [-StorageLogicalUnit] <StorageLogicalUnit[]> -JobGroup <Guid> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

Parameter Set: RegisterToCluster

```
Register-SCStorageLogicalUnit [-StorageLogicalUnit] <StorageLogicalUnit[]> -VMHostCluster <HostCluster> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

Parameter Set: RegisterToHost

```
Register-SCStorageLogicalUnit [-StorageLogicalUnit] <StorageLogicalUnit[]> -VMHost <Host> [-JobGroup <Guid> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

Parameter Set: RegisterToInitiators

```
Register-SCStorageLogicalUnit [-StorageLogicalUnit] <StorageLogicalUnit[]> -StorageInitiators <String[]> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-StorageEndpoints <String[]> ] [-StorageGroupName <String> ] [ <CommonParameters>]
```

### Detailed Description

The Register-SCStorageLogicalUnit associates a logical unit with a virtual machine host. The logical unit appears in the operating system as a disk.

### Parameters

#### **-JobGroup<Guid>**

Specifies an identifier for a series of commands that will run as a set just before the final command that includes the same job group identifier runs.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

---

|                             |       |
|-----------------------------|-------|
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

---

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-StorageEndpoints<String[]>**

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-StorageGroupName<String>**

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-StorageInitiators<String[]>**

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | true  |
| Position? | named |

|                             |       |
|-----------------------------|-------|
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-StorageLogicalUnit<StorageLogicalUnit[]>**

Specifies a storage logical unit object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-VMHost<Host>**

Specifies a virtual machine host object. VMM supports Hyper-V hosts, VMware ESX hosts, and Citrix XenServer hosts.

For more information about each type of host, type: "Get-Help Add-SCVMHost -detailed". See the examples for a specific cmdlet to determine how that cmdlet uses this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMHostCluster<HostCluster>**

Specifies a VMM host cluster object.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|



---

|                             |       |
|-----------------------------|-------|
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **StorageDisk**

## Examples

### 1: Register a logical unit with a host.

The first command gets the host object named VMHost01 and stores the object in the \$VMHost variable.

The second command gets the storage logical unit object named LUN01 and stores the object in the \$LU variable.

The last command registers LUN01 with VMHost01.

```
PS C:\> $VMHost = Get-SCVMHost -ComputerName "VMHost01"
```

```
PS C:\> $LU = Get-SCStorageLogicalUnit -Name "LUN01"
```

```
PS C:\> Register-SCStorageLogicalUnit -StorageLogicalUnit $LU -VMHost $VMHost
```

## Related topics

[Get-SCStorageLogicalUnit](#)

[New-SCStorageLogicalUnit](#)

[Remove-SCStorageLogicalUnit](#)

[Set-SCStorageLogicalUnit](#)

[Unregister-SCStorageLogicalUnit](#)

---

# Register-SCVirtualMachine

---

## Register-SCVirtualMachine

Registers an existing virtual machine with VMM that is currently not registered with the virtualization platform of any host managed by VMM and is not stored in the VMM library.

### Syntax

Parameter Set: Default

```
Register-SCVirtualMachine [-Path] <String> [-VMHost] <Host> [-JobGroup <Guid> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

### Detailed Description

The Register-SCVirtualMachine cmdlet registers an existing virtual machine with Virtual Machine Manager (VMM) that is not currently registered with the virtualization platform (Hyper-V, VMware, or XenServer) of any host managed by VMM, and is not stored in the VMM library. If virtual machine files are stored in the VMM library, you do not need to register the virtual machine before you deploy it on a host.

The configuration files for the virtual machine that you want to register must be stored either in the file system on the host on which you want to deploy the virtual machine or stored on shared storage (such as a SAN) available to this host:

- HYPER-V HOST – You can register a virtual machine for a Hyper-V host

if the configuration files for that virtual machine are stored in a folder on the host's file system or on shared storage, and an export of the virtual machine was created using the "Export" function in the Hyper-V Manager console.

The path must specify a folder. Example: -Path "D:\HyperVFolderForVMs"

- VMWARE ESX HOST - You can register a virtual machine for a VMware ESX host if the VMware virtual machine configuration file (a .vmx file) is stored on the host's file system or on shared storage. No separate "export" step is required.

The path must specify the folder and the configuration file. Example: -Path [Datastore]VMwareFolderForVMs\VM.vmx

- CITRIX XENSERVEN HOST - This command does not apply to virtual machines on a XenServer host.

---

## Parameters

### **-JobGroup<Guid>**

Specifies an identifier for a series of commands that will run as a set just before the final command that includes the same job group identifier runs.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Path<String>**

Specifies the destination path for the operation.

Example formats:

Local path -Path "F:\"

UNC path -Path "\\Library\Templates"

Volume GUID path -Path "\\?\Volume{4703c1ea-8ae7-11db-b473-00123f7603e3}\"

VMware ESX path -Path "[storage1]\MyVMwareFolderForVMs\MyVM.vmx"

Citrix XenServer path - Path "Local storage[99b6212f-b63d-c676-25f9-d6c460992de7]"

Wildcards are supported for "Get" cmdlets and when you specify the UNC path:

Example format:

---

UNC path     -Path "\\VMHostServer\MyVMs\\*VM\*"

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | 1     |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-VMHost<Host>**

Specifies a virtual machine host object. VMM supports Hyper-V hosts, VMware ESX hosts, and Citrix XenServer hosts.

For more information about each type of host, type: "Get-Help Add-SCVMHost -detailed". See the examples for a specific cmdlet to determine how that cmdlet uses this parameter.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **VirtualMachine**

## **Examples**

### **1: Register an existing virtual machine on a Hyper-V host.**

The first command gets the Hyper-V host object named HVHost02 and stores the object in the \$VMHost variable.

The second command adds the existing virtual machine on HVHost02 to VMM by specifying the path to the folder that contains the virtual machine configuration file.

```
PS C:\> $VMHost = Get-SCVMHost -ComputerName "HyperVHost01"
```

```
PS C:\> Register-SCVirtualMachine -VMHost $VMHost -Path "D:\HyperVFolderForVMs"
```

### **2: Register an existing virtual machine on a VMware ESX host.**

The first command gets the object that represents a VMware ESX host called ESXHost03 and stores the host object in \$VMHost.

---

The last command adds an existing virtual machine on ESXHost03 to VMM by specifying the path to the virtual machine's virtual machine configuration file (MyVM.vmx).

```
PS C:\> $VMHost = Get-VMHost -ComputerName "ESXHost03"
```

```
PS C:\> Register-VM -VMHost $VMHost -Path "[storage1]\VMwareFolderForVMs\MyVM.vmx"
```

## Related topics

[Get-SCVMHost](#)

---

# Register-SCVMHost

---

## Register-SCVMHost

Associates a VMware ESX host with VMM as a virtual machine host and specifies credentials to manage the host.

### Syntax

Parameter Set: Default

```
Register-SCVMHost [-VMHost] <Host> [-Certificate <ClientCertificate> ] [-Credential <VMMCredential> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-SshPublicKey <ClientSshPublicKey> ] [-SshPublicKeyFile <String> ] [-SshTcpPort <UInt32> ] [-TCPPort <UInt32> ] [ <CommonParameters>]
```

### Detailed Description

The Register-SCVMHost cmdlet associates a VMware ESX host with Virtual Machine Manager (VMM) as a virtual machine host and specifies the credentials to use with this ESX host.

### Parameters

#### **-Certificate<ClientCertificate>**

Specifies a security certificate object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-Credential<VMMSCredential>**

Specifies a credential object or, for some cmdlets, a Run As account object that contains the user name and password of an account that has permission to perform this action. Or, in the case of Restart-SCJob, has permission to complete a restarted task.

For more information about the PSCredential object, type: "Get-Help Get-Credential".

For more information about Run As accounts, type: "Get-Help New-SCRunAsAccount".

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                        |       |
|------------------------|-------|
| Aliases                | none  |
| Required?              | false |
| Position?              | named |
| Default Value          | none  |
| Accept Pipeline Input? | false |



---

|                             |       |
|-----------------------------|-------|
| Accept Wildcard Characters? | false |
|-----------------------------|-------|

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-SshPublicKey<ClientSshPublicKey>**

Specifies the public key used by Secure Shell (SSH) communications.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-SshPublicKeyFile<String>**

Specifies the path to the public key file for establishing a secured SSH channel with the target hosts.

|                        |       |
|------------------------|-------|
| Aliases                | none  |
| Required?              | false |
| Position?              | named |
| Default Value          | none  |
| Accept Pipeline Input? | false |

---

|                             |       |
|-----------------------------|-------|
| Accept Wildcard Characters? | false |
|-----------------------------|-------|

### **-SshTcpPort<UInt32>**

Specifies the TCP port number used by the Secure Shell (SSH) protocol.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-TCPPort<UInt32>**

Specifies a numeric value that represents a TCP port.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMHost<Host>**

Specifies a virtual machine host object. VMM supports Hyper-V hosts, VMware ESX hosts, and Citrix XenServer hosts.

For more information about each type of host, type: "Get-Help Add-SCVMHost -detailed". See the examples for a specific cmdlet to determine how that cmdlet uses this parameter.

|           |      |
|-----------|------|
| Aliases   | none |
| Required? | true |
| Position? | 1    |

---

|                             |                |
|-----------------------------|----------------|
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **Host**

## Examples

### 1: Set the credentials for a specific VMware ESX host.

The first command gets the Run As account object named ESX Host Computer Acct and stores the object in the \$RunAsAccount variable.

The second command gets the ESX host object named ESXHost02 and stores the object in the \$ESXHost variable.

The last command associates the VMware ESX host with VMM as a managed host, and specifies that the Run As account stored in \$Credential should be used to access ESXHost02.

```
PS C:\> $RunAsAccount = Get-SCRunAsAccount -Name "ESX Host Computer Acct"
```

```
PS C:\> $ESXHost = Get-SCVMHost -ComputerName "ESXHost02.Contoso.com"
```

```
PS C:\> Register-SCVMHost -VMHost $ESXHost -Credential $RunAsAccount
```

### 2: Set the credentials and certificate for a specific VMware ESX host.

The first command gets the Run As account object named ESX Host Computer Acct and stores the object in the \$RunAsAccount variable. The required credentials for this operation are either a root account (root/<password>) or the account for the VMware delegated administrator defined earlier in VirtualCenter Server for this ESX host..

The second command gets the the VMware ESX host object named ESXHost03 and stores the object in the \$ESXHost variable.

The third command uses the Get-SCCertificate cmdlet to get the certificate object from ESXHost02 and stores the object in the \$Cert variable.

The last command associates this VMware ESX Server with VMM as a managed host and specifies that the credentials used to access ESXHost02 are those stored in \$RunAsAccount.

```
PS C:\> $RunAsAccount = Get-SCRunAsAccount -Name "ESX Host Computer Acct"
```

---

```
PS C:\> $ESXHost = Get-SCVMHost -ComputerName "ESXHost03.contoso.com"  
PS C:\> $Cert = Get-SCCertificate -ComputerName "ESXHost03.contoso.com"  
PS C:\> Register-SCVMHost -VMHost $ESXHost -Credential $RunAsAccount -Certificate $Cert
```

## Related topics

[Get-SCVirtualizationManager](#)

[Add-SCVirtualizationManager](#)

[Get-SCCertificate](#)

[Set-SCVirtualizationManager](#)

---

# Register-SCVMMAccessLicense

---

## Register-SCVMMAccessLicense

Registers VMM using the provided product key.

### Syntax

Parameter Set: Default

```
Register-SCVMMAccessLicense -ProductKey <String> [-AcceptEULA] [-JobVariable <String> ] [-  
PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [  
<CommonParameters>]
```

### Detailed Description

The Register-SVMMAccessLicense cmdlet registers Virtual Machine Manager (VMM), using the valid product key provided by the user. This operation upgrades the evaluation version of VMM to a registered version.

### Parameters

#### -AcceptEULA

Indicates that the Microsoft Software License Terms are accepted.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### -JobVariable<String>

Specifies that job progress is tracked and stored in the variable named by this parameter.

---

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-ProductKey<String>**

Specifies a product key. The product key is a 25-digit number that identifies the product license. A product key can be used to register VMM or an operating system to be installed on a virtual machine or host.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## **Examples**

### **1: Upgrade from an evaluation version to a registered version of VMM by providing a product key.**

This command registers this VMM server to which the user is currently connected, including all of its nodes if it is a highly available VMM server, with the provided product key.

```
PS C:\> Register-SCVMMAccessLicense -ProductKey "XXXXX-XXXXX-XXXXX-XXXXX-XXXXX"
```

## **Related topics**

[Get-SCVMMAccessLicense](#)

---

# Register-SCVMMManagedComputer

---

## Register-SCVMMManagedComputer

Reassociates a managed computer on which VMM agent software is installed with a different VMM management server.

### Syntax

Parameter Set: Default

```
Register-SCVMMManagedComputer [-VMMManagedComputer] <VMMManagedComputer> -Credential  
<VMMServerCredential> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [  
<CommonParameters>]
```

### Detailed Description

The Register-SCVMMManagedComputer cmdlet reassociates a managed computer on which Virtual Machine Manager (VMM) agent software is installed with a different VMM management server.

When you initially add a host or library server to VMM, the host or library server is "associated" with the VMM management server that provides the VMM database that you added the host or library server to. The VMM database might be installed in a Microsoft SQL Server database on the VMM management server itself or on a remote computer running SQL Server.

After a host or library server is added to (and therefore associated with) a VMM management server, it cannot communicate with any other VMM management server. However, you can reassociate it with a different VMM management server, as described in the following scenarios.

#### SCENARIO 1: DISASTER RECOVERY

-----

In this scenario, VMMServerA fails, or the VMM service running on VMMServerA fails. You might already have VMMServerB available as a backup VMM management server. If not, you can install the VMM service on VMMServerB. At this point, the VMM database might be on VMMServerB, or, if you keep the database on a separate SQL Server, you can now point VMMServerB to the VMM database on that SQL Server.

However, although you now have a functioning VMM management server (VMMServerB) and database, hosts and library servers that were managed by VMMServerA are still configured to communicate with VMMServerA. VMMServerB recognizes these managed computers, but they are in an "Access Denied" state. At this point, you can use Reassociate-SCVMMManagedComputer to reassociate computers that were managed by VMMServerA with VMMServerB.

#### SCENARIO 2: RE-ORGANIZING SERVER GROUPINGS

-----



---

In this scenario, VMMServerA and VMMServerB are two existing VMM management servers that manage different sets of hosts and library servers. If, for example, VMHost01 is currently managed by VMMServerA, you can add VMHost01 to VMMServerB by using the Add-SCVMHost cmdlet with the Reassociate parameter. If you do this, the state of VMHost01 on VMMServerA is now "Access Denied" and its state on VMMServerB is "Responding." VMHost01 is now managed by VMMServerB, so you can remove it from VMMServerA.

You can also use Reassociate-SCVMMManagedComputer to reassociate Host01 with VMMServerA.

## Parameters

### **-Credential<VMMCredential>**

Specifies a credential object or, for some cmdlets, a Run As account object that contains the user name and password of an account that has permission to perform this action. Or, in the case of Restart-SCJob, has permission to complete a restarted task.

For more information about the PSCredential object, type: "Get-Help Get-Credential".

For more information about Run As accounts, type: "Get-Help New-SCRunAsAccount".

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VMMManagedComputer<VMMManagedComputer>**

Specifies a computer object that is managed by VMM.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **VMMManagedComputer**

## Examples

### 1: Reassociate all unassociated managed computers with a specific VMM server.

The first command connects to VMMServer01 in the Contoso.com domain.

The second command uses Get-Credential to prompt you to supply a user name and password and stores your credentials in the \$Credential variable. The required credentials for this operation are a domain account with administrator rights on the host server that you want to reassociate with a specific VMM server and the password for that account.

The last command gets all managed computers from VMMServer01 and selects only those objects that are in the Not Responding state. Then, it passes these objects to the Register-SCVMMManagedComputer cmdlet which changes the association of the objects to VMMServer01. As this command is processed, \$Credential provides your credentials to Register-SCVMMManagedComputer.

```
PS C:\> Get-VMMServer -ComputerName "VMMServer01.Contoso.com"
```

```
PS C:\> $Credential = Get-Credential
```

```
PS C:\> Get-VMMManagedComputer | where {$_.State -eq "NotResponding"} | Register-SCVMMManagedComputer -Credential $Credential
```

## Related topics

[Get-SCVMMManagedComputer](#)

[Update-SCVMMManagedComputer](#)

---

# Remove-SCApplicationDeployment

---

## Remove-SCApplicationDeployment

Deletes an application deployment from an application profile.

### Syntax

Parameter Set: Default

```
Remove-SCApplicationDeployment [-ApplicationDeployment] <ApplicationDeployment> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

### Detailed Description

The Remove-SCApplicationDeployment cmdlet deletes an application deployment from an application profile.

### Parameters

#### **-ApplicationDeployment<ApplicationDeployment>**

Specifies an application deployment object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

---

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

---

## Examples

### 1: Remove all of the application deployments from an application profile

The first command gets the application profile object named SvcWebAppProfile01 and stores the object in the \$AppProfile variable.

The second command gets all application deployment objects for the application profile stored in \$AppProfile and stores the objects in the \$AppDeployment array.

The last command uses the pipeline operator to pass each of the objects stored in the \$AppDeployment array to the Remove-SCApplicationDeployment cmdlet, which deletes the application deployment objects.

```
PS C:\> $AppProfile = Get-SCApplicationProfile -Name "SvcWebAppProfile01"
```

```
PS C:\> $AppDeployment = Get-SCApplicationDeployment -ApplicationProfile $AppProfile
```

```
PS C:\> $AppDeployment | Remove-SCApplicationDeployment
```

### Related topics

[Add-SCApplicationDeployment](#)

[Get-SCApplicationDeployment](#)

[Get-SCApplicationProfile](#)

[Set-SCApplicationDeployment](#)

---

# Remove-SCApplicationHostTemplate

---

## Remove-SCApplicationHostTemplate

Removes an application host template from a service template.

### Syntax

Parameter Set: Default

```
Remove-SCApplicationHostTemplate [-ApplicationHostTemplate] <ApplicationHostTemplate> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

### Detailed Description

The Remove-SCApplicationHostTemplate removes an application host template from a service template.

### Parameters

#### **-ApplicationHostTemplate<ApplicationHostTemplate>**

Specifies an application host template object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

---

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Confirm**

Prompts you for confirmation before executing the command.



|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Examples**

### **1: Remove an application host template from a service template.**

The first command gets the service template object named ServiceTemplate01 and stores the object in the \$ServiceTemplate variable.

The second command gets the application host template object for the service template in \$ServiceTemplate and stores the object in the \$AppHostTemplate variable.

The last command removes the application host template in \$AppHostTemplate.

```
PS C:\> $ServiceTemplate = Get-SCServiceTemplate -Name "ServiceTemplate01"
```

```
PS C:\> $AppHostTemplate = Get-SCApplicationHostTemplate -ServiceTemplate $ServiceTemplate
```

```
PS C:\> Remove-SCApplicationHostTemplate -ApplicationHostTemplate $AppHostTemplate
```

---

## Related topics

[Add-SCApplicationHostTemplate](#)

[Get-SCApplicationHostTemplate](#)

[Get-SCServiceTemplate](#)

[Set-SCApplicationHostTemplate](#)

---

# Remove-SCApplicationPackage

---

## Remove-SCApplicationPackage

Removes an application package from the VMM library.

### Syntax

Parameter Set: Default

```
Remove-SCApplicationPackage [-ApplicationPackage] <ApplicationPackage> [-Force] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-Confirm] [-WhatIf] [<CommonParameters>]
```

### Detailed Description

The Remove-SCApplicationPackage cmdlet deletes an application package from the VMM library.

### Parameters

#### **-ApplicationPackage<ApplicationPackage>**

Specifies an application package object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

#### **-Force**

Forces the operation to complete.

For example:

```
- Remove-SCSCVMHost -Force
```

---

Forces the removal of a host object from the VMM database.

- Stop-SCVirtualMachine -Force

Stops a virtual machine.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-Confirm**

Prompts you for confirmation before executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Examples

### 1: Remove a specific application profile.

The first command gets the application package object named AppPackage01 with the release value of Beta and stores the object in the \$AppPackage variable.

The second command removes the application package stored in \$AppPackage.

```
PS C:\> $AppPackage = Get-SCApplicationPackage -Name "AppPackage01" -Release "Beta"
```

```
PS C:\> Remove-SCApplicationPackage -ApplicationPackage $AppPackage
```

## Related topics

[Get-SCApplicationPackage](#)

[Set-SCApplicationPackage](#)

---

# Remove-SCApplicationProfile

---

## Remove-SCApplicationProfile

Deletes an application profile.

### Syntax

Parameter Set: Default

```
Remove-SCApplicationProfile [-ApplicationProfile] <ApplicationProfile> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

### Detailed Description

The Remove-SCApplicationProfile cmdlet deletes an application profile from Virtual Machine Manager (VMM).

### Parameters

#### **-ApplicationProfile<ApplicationProfile>**

Specifies an application profile object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

---

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Confirm**

Prompts you for confirmation before executing the command.



|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Examples**

### **1: Remove a specific application profile.**

The first command gets the application profile object named SvcWebAppProfile01 and stores the object in the \$AppProfile variable.

The second command removes the application profile stored in \$AppProfile. A confirmation prompt is displayed before the profile is removed.

```
PS C:\> $AppProfile = Get-SCApplicationProfile -Name "SvcWebAppProfile01"
```

```
PS C:\> Remove-SCApplicationProfile -ApplicationProfile $AppProfile -Confirm
```

## **Related topics**

[Get-SCApplicationProfile](#)

[New-SCApplicationProfile](#)

---

[Set-SCApplicationProfile](#)

---

# Remove-SCBaseline

---

## Remove-SCBaseline

Removes a baseline from VMM.

### Syntax

Parameter Set: Default

```
Remove-SCBaseline [-Baseline] <Baseline> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

### Detailed Description

The Remove-SCBaseline cmdlet removes a baseline and its scope assignments from Virtual Machine Manager (VMM). Compliance is no longer graded for this baseline.

For more information about baselines, type: "Get-Help New-SCBaseline".

### Parameters

#### **-Baseline<Baseline>**

Specifies a VMM baseline object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

---

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-Confirm**

Prompts you for confirmation before executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- 
- **Baseline**

## Examples

### 1: Remove a baseline.

The first command gets the baseline object named Security Baseline and stores the object in the \$Baseline variable.

The second command removes the baseline stored in \$Baseline (Security Baseline).

```
PS C:\> $Baseline = Get-SCBaseline -Name "Security Baseline"
```

```
PS C:\> Remove-SCBaseline -Baseline $Baseline
```

## Related topics

[Get-SCBaseline](#)

[New-SCBaseline](#)

[Set-SCBaseline](#)

---

# Remove-SCCapabilityProfile

---

## Remove-SCCapabilityProfile

Deletes a capability profile from VMM.

### Syntax

Parameter Set: Default

```
Remove-SCCapabilityProfile [-CapabilityProfile] <CapabilityProfile> [-JobVariable <String> ]  
[-PROTipID <Guid> ] [-RunAsynchronously] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The Remove-SCCapabilityProfile cmdlet deletes a capability profile from Virtual Machine Manager (VMM).

### Parameters

#### **-CapabilityProfile<CapabilityProfile>**

Specifies a capability profile object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

---

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Confirm**

Prompts you for confirmation before executing the command.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|



|                             |       |
|-----------------------------|-------|
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## -WhatIf

Describes what would happen if you executed the command without actually executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Notes

- Requires a VMM capability profile object, which can be retrieved using the Get-SCCapabilityProfile cmdlet.

## Examples

### 1: Delete a specific capability profile.

The first command gets the capability profile object named CapabilityProf01 and stores the object in the \$CapabilityProfile variable.

The last command deletes the capability profile stored in \$CapabilityProfile (CapabilityProf01), prompting you for confirmation prior to performing the operation.

```
PS C:\> $CapabilityProfile = Get-SCCapabilityProfile -Name "CapabilityProf01"
```

```
PS C:\> Remove-SCCapabilityProfile -CapabilityProfile $CapabilityProfile -Confirm
```

---

## Related topics

[Get-SCCapabilityProfile](#)

[New-SCCapabilityProfile](#)

[Set-SCCapabilityProfile](#)

[Test-SCCapabilityProfile](#)

---

# Remove-SCCloud

---

## Remove-SCCloud

Deletes a private cloud from VMM.

### Syntax

Parameter Set: Default

```
Remove-SCCloud [-Cloud] <Cloud> [-JobGroup <Guid> ] [-JobVariable <String> ] [-PROTipID  
<Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

### Detailed Description

The Remove-SCCloud cmdlet deletes a private cloud from Virtual Machine Manager (VMM).

### Parameters

#### **-Cloud<Cloud>**

Specifies a private cloud object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

#### **-JobGroup<Guid>**

Specifies an identifier for a series of commands that will run as a set just before the final command that includes the same job group identifier runs.

---

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-Confirm**

Prompts you for confirmation before executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

---

|                             |       |
|-----------------------------|-------|
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## Notes

- Requires a VMM private cloud object, which can be retrieved by using the Get-SCCloud cmdlet.

## Examples

### 1: Delete a specified private cloud.

This command gets the private cloud object named Cloud01 on VMMServer01 and uses the pipeline operator to pass the private cloud object to Remove-SCCloud which prompts the user to confirm the action before deleting the private cloud.

```
PS C:\> Get-SCCloud -VMMServer "VMMServer01.Contoso.com" -Name "Cloud01" | Remove-SCCloud -Confirm
```

## Related topics

[Get-SCCloud](#)

[New-SCCloud](#)

[Set-SCCloud](#)

---

# Remove-SCComputerConfiguration

---

## Remove-SCComputerConfiguration

Removes a machine configuration object from VMM.

### Syntax

Parameter Set: Default

```
Remove-SCComputerConfiguration [-ComputerConfiguration] <MachineConfiguration> [-Credential <PSCredential> ] [-Force] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

### Detailed Description

The Remove-SCComputerConfiguration cmdlet removes one or more computer configuration objects from the Virtual Machine Manager (VMM) database and removes the VMM P2V agent from the physical source computer (if the agent is still installed).

This cmdlet returns the object upon success (with the property MarkedForDeletion set to TRUE) or returns an error message upon failure.

### Parameters

#### **-ComputerConfiguration<MachineConfiguration>**

Specifies a computer configuration object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

---

## **-Credential<PSCredential>**

Specifies a credential object or, for some cmdlets, a Run As account object that contains the user name and password of an account that has permission to perform this action. Or, in the case of Restart-SCJob, has permission to complete a restarted task.

For more information about the PSCredential object, type: "Get-Help Get-Credential".

For more information about Run As accounts, type: "Get-Help New-SCRunAsAccount".

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-Force**

Forces the operation to complete.

For example:

- Remove-SCSCVMHost -Force

Forces the removal of a host object from the VMM database.

- Stop-SCVirtualMachine -Force

Stops a virtual machine.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.



|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

---

## Notes

- Requires a VMM computer configuration object, which can be retrieved by using the Get-SCComputerConfiguration cmdlet.

## Examples

### 1: Remove a specific computer configuration from the VMM database.

The first command gets the computer configuration object named P2VSource01.Contoso.com from the VMM database and stores the object in the \$ComputerConfig variable.

The second command removes the machine configuration object for P2VSource01.Contoso.com from the VMM database by using the -Force parameter.

```
PS C:\> $ComputerConfig = Get-SCComputerConfiguration | where { $_.Name -eq "P2VSource01.Contoso.com" }
```

```
PS C:\> Remove-SCComputerConfiguration -ComputerConfiguration $ComputerConfig -Force
```

### 2: Remove a P2V agent from a specific source computer.

The first command uses Get-Credential to prompt you to supply a user name and password and stores your credentials in \$Credential. The required credentials for this operation are either a local Administrator account or a domain account with administrator rights on the computer from which you want to gather information.

The command gets the computer configuration object named P2VSource01.Contoso.com and stores the object in the \$ComputerConfig variable.

The last command removes the P2V agent from P2VSource01.Contoso.com.

```
PS C:\> $Credential = Get-Credential
```

```
PS C:\> $ComputerConfig = Get-SCComputerConfiguration | where { $_.Name -eq "P2VSource01.Contoso.com" }
```

```
PS C:\> Remove-SCComputerConfiguration -ComputerConfiguration $ComputerConfig -Credential $Credential
```

## Related topics

[Get-SCComputerConfiguration](#)

[New-SCComputerConfiguration](#)

---

# Remove-SCComputerTierTemplate

---

## Remove-SCComputerTierTemplate

Removes a computer tier template from a service template.

### Syntax

Parameter Set: Default

```
Remove-SCComputerTierTemplate -ComputerTierTemplate <ComputerTierTemplate> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

### Detailed Description

The Remove-SCComputerTierTemplate cmdlet removes a computer tier template from a service template.

### Parameters

#### **-ComputerTierTemplate<ComputerTierTemplate>**

Specifies a computer tier template object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

---

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Confirm**

Prompts you for confirmation before executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Examples**

### **1: Remove a computer tier template from a service template.**

The first command gets the service template object named ServiceTemplate01 and stores the object in the \$ServiceTemplate variable.

The second command gets the computer tier template for the service template stored in \$ServiceTemplate.

The last command removes the computer tier template stored in \$TierTemplate after prompting the user for confirmation.

```
PS C:\> $ServiceTemplate = Get-SCServiceTemplate -Name "ServiceTemplate01"
PS C:\> $TierTemplate = Get-SCComputerTierTemplate -ServiceTemplate $ServiceTemplate
PS C:\> Remove-SCComputerTierTemplate -ComputerTierTemplate $TierTemplate -Confirm
```

---

## Related topics

[Add-SCComputerTierTemplate](#)

[Get-SCComputerTierTemplate](#)

[Get-SCServiceTemplate](#)

[Set-SCComputerTierTemplate](#)

---

# Remove-SCCustomPlacementRule

---

## Remove-SCCustomPlacementRule

Deletes a custom placement rule from a placement configuration.

### Syntax

Parameter Set: FromPlacementConfiguration

```
Remove-SCCustomPlacementRule -CustomPropertyName <String> -PlacementConfiguration  
<PlacementConfigurationSettings> [-JobVariable <String> ] [-PROTipID <Guid> ] [-  
RunAsynchronously] [-VMMServer <ServerConnection> ] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

### Detailed Description

The Remove-SCCustomPlacementRule cmdlet deletes a custom placement rule from a placement configuration for a host group.

### Parameters

#### **-CustomPropertyName<String>**

Specifies the name for a custom property.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PlacementConfiguration<PlacementConfigurationSettings>**

Specifies a placement configuration object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.



|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Confirm**

Prompts you for confirmation before executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

---

|                             |       |
|-----------------------------|-------|
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## Examples

### 1: Remove a custom placement rule from a placement configuration.

The first command gets the host group object named HostGroup01 and stores the object in the \$HostGroup variable.

The second command gets the placement configuration object for the host group stored in \$HostGroup and places the object in the \$PlacementConfig variable.

The last command uses the pipeline operator to pass the placement configuration stored in \$PlacementConfig to the Remove-SCCustomPlacementRule cmdlet. Remove-SCCustomPlacementRule removes the custom placement rule named Cost Center from the placement configuration for HostGroup01 after prompting the user for confirmation.

```
PS C:\> $HostGroup = Get-SCVMHostGroup "HostGroup01"
```

```
PS C:\> $PlacementConfig = Get-SCPlacementConfiguration -VMHostGroup $HostGroup
```

```
PS C:\> $PlacementConfig | Remove-SCCustomPlacementRule -CustomPropertyName "Cost Center" -Confirm
```

## Related topics

[Get-SCCustomPlacementRule](#)

[Get-SCPlacementConfiguration](#)

[Get-SCVMHostGroup](#)

---

# Remove-SCCustomProperty

---

## Remove-SCCustomProperty

Removes a custom property definition from the VMM database.

### Syntax

Parameter Set: Default

```
Remove-SCCustomProperty -CustomProperty <CustomProperty> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The Remove-SCCustomProperty cmdlet removes a custom property definition from the Virtual Machine Manager (VMM) database.

### Parameters

#### **-CustomProperty<CustomProperty>**

Specifies a custom property object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Confirm**

Prompts you for confirmation before executing the command.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

---

|                             |       |
|-----------------------------|-------|
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Notes**

- Requires a VMM custom property object, which can be retrieved by using the Get-SCCustomProperty cmdlet.

## **Examples**

### **1: Remove a custom property definition.**

The first command gets the custom property object named Cost Center and stores the object in the \$CustomProp variable.

The second command removes the custom property object stored in \$CustomProp.

```
PS C:\> $CustomProp = Get-SCCustomProperty -Name "Cost Center"
```

```
PS C:\> Remove-SCCustomProperty -CustomProperty $CustomProp
```

---

## Related topics

[Get-SCCustomProperty](#)

[New-SCCustomProperty](#)

[Set-SCCustomProperty](#)

---

# Remove-SCCustomPropertyValue

---

## Remove-SCCustomPropertyValue

Removes the value from a custom property.

### Syntax

Parameter Set: CustomPropertyValue

```
Remove-SCCustomPropertyValue -CustomPropertyValue <CustomPropertyValue> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

Parameter Set: JobGroup

```
Remove-SCCustomPropertyValue -CustomProperty <CustomProperty> -JobGroup <Guid> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

### Detailed Description

The Remove-SCCustomPropertyValue cmdlet removes the value from a custom property.

### Parameters

#### **-CustomProperty<CustomProperty>**

Specifies a custom property object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### **-CustomPropertyValue<CustomPropertyValue>**

Specifies a custom property value object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-JobGroup<Guid>**

Specifies an identifier for a series of commands that will run as a set just before the final command that includes the same job group identifier runs.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |



---

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## **Examples**

### **1: Remove a custom property value from a virtual machine.**

The first command gets the virtual machine object named VM01 and stores the object in the \$VM variable.

The second command gets the custom property object named Cost Center and stores the object in the \$CustomProp variable.

The third command retrieves the value for the custom property stored in \$CustomProp (Cost Center) for the virtual machine stored in \$VM (VM01) and stores the value in the \$CustomPropValue variable.

---

The last command removes the custom property value stored in \$CustomPropValue.

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM01"
```

```
PS C:\> $CustomProp = Get-SCCustomProperty -Name "Cost Center"
```

```
PS C:\> $CustomPropValue = Get-SCCustomPropertyValue -InputObject $VM -CustomProperty  
$CustomProp
```

```
PS C:\> Remove-SCCustomPropertyValue -CustomPropertyValue $CustomPropValue
```

## Related topics

[Get-SCCustomProperty](#)

[Get-SCCustomPropertyValue](#)

[Set-SCCustomPropertyValue](#)

---

# Remove-SCCustomResource

---

## Remove-SCCustomResource

Removes a custom resource from the VMM library.

### Syntax

Parameter Set: Default

```
Remove-SCCustomResource [-CustomResource] <CustomResource> [-Force] [-JobVariable <String> ]  
[-PROTipID <Guid> ] [-RunAsynchronously] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The Remove-SCCustomResource cmdlet removes a custom resource from the VMM library. For more information about custom resources, type: "Get-Help Set-SCCustomResource".

### Parameters

#### **-CustomResource<CustomResource>**

Specifies a custom resource object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

#### **-Force**

Forces the operation to complete.

For example:

```
- Remove-SCSCVMHost -Force
```

---

Forces the removal of a host object from the VMM database.

- Stop-SCVirtualMachine -Force

Stops a virtual machine.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-Confirm**

Prompts you for confirmation before executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- CustomResource

## Examples

### 1: Remove a custom resource.

The first command gets the custom resource object named Folder.CR on LibraryServer01 from the VMM library on VMMServer01 and then stores the object in the \$CR variable.

The second command removes the custom resource stored in \$CR from the VMM library.

```
PS C:\> $CR = Get-SCCustomResource -VMMServer "VMMServer01.Contoso.com" | where { $_.Name -eq "Folder.CR" -and $_.LibraryServer.Name -eq "LibraryServer01.Contoso.com" }
```

```
PS C:\> Remove-SCCustomResource -CustomResource $CR
```

## Related topics

[Get-SCCustomResource](#)

[Set-SCCustomResource](#)

---

# Remove-SCDriverPackage

---

## Remove-SCDriverPackage

Removes a driver package object from the VMM library.

### Syntax

Parameter Set: Default

```
Remove-SCDriverPackage [-DriverPackage] <DriverPackage> [-Force] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The Remove-SCDriverPackage cmdlet removes a driver package object from the VMM library.

### Parameters

#### **-DriverPackage<DriverPackage>**

Specifies a driver package object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

#### **-Force**

Forces the operation to complete.

For example:

```
- Remove-SCSCVMHost -Force
```

Forces the removal of a host object from the VMM database.

---

- Stop-SCVirtualMachine -Force  
Stops a virtual machine.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |



---

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-Confirm**

Prompts you for confirmation before executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Examples

### 1: Remove a specific driver package.

This command removes the driver package object with the Plug and Play ID of "DRIVER" from the VMM library.

```
PS C:\> Get-SCDriverPackage -PnPID "DRIVER" | Remove-SCDriverPackage -Confirm
```

## Related topics

[Get-SCDriverPackage](#)

[Set-SCDriverPackage](#)

---

# Remove-SCGuestOSProfile

---

## Remove-SCGuestOSProfile

Removes a guest operating system profile object from VMM.

### Syntax

Parameter Set: Default

```
Remove-SCGuestOSProfile [-GuestOSProfile] <GuestOSProfile> [-Force] [-JobVariable <String> ]  
[-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

### Detailed Description

The Remove-SCGuestOSProfile cmdlet removes one or more guest operating system profile objects from the VMM library.

This cmdlet returns the object upon success (with the property MarkedForDeletion set to True) or returns an error message upon failure.

### Parameters

#### -Force

Forces the operation to complete.

For example:

- Remove-SCSCVMHost -Force

Forces the removal of a host object from the VMM database.

- Stop-SCVirtualMachine -Force

Stops a virtual machine.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-GuestOSProfile<GuestOSProfile>**

Specifies a guest operating system profile object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## -RunAsynchronously

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Notes

- Requires a VMM guest operating system profile object, which can be retrieved by using the Get-SCGuestOSProfile cmdlet.

## Examples

### 1: Remove a specific guest operating system profile from the library.

The first command gets the guest OS profile object named NewOSProfile01 and stores the object in the \$OSProfile variable.

The second command removes the guest OS profile stored in \$OSProfile, prompting for confirmation before completing the operation.

```
PS C:\> $OSProfile = Get-SCGuestOSProfile -Name "NewOSProfile01"
PS C:\> Remove-SCGuestOSProfile -GuestOSProfile $OSProfile -Confirm
```

### 2: Remove all operating system profiles without being prompted to confirm each deletion.

The first command gets all operating system profile objects from VMMServer01 and stores the objects in the \$OSProfiles object array.

---

The second command passes each object in \$OSProfiles to the Remove-OSProfile cmdlet, which removes each of the guest OS profile objects from the VMM library.

```
PS C:\> $OSProfiles = Get-SCGuestOSProfile -VMMServer "VMMServer01.Contoso.com"
```

```
PS C:\> $OSProfiles | Remove-SCGuestOSProfile
```

## Related topics

[Get-SCGuestOSProfile](#)

[New-SCGuestOSProfile](#)

[Set-SCGuestOSProfile](#)

---

# Remove-SCHardwareProfile

---

## Remove-SCHardwareProfile

Removes a hardware profile object from the VMM library.

### Syntax

Parameter Set: Default

```
Remove-SCHardwareProfile [-HardwareProfile] <HardwareProfile> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

### Detailed Description

The Remove-SCHardwareProfile cmdlet removes one or more hardware profile objects from the VMM library.

This cmdlet returns the object upon success (with the property MarkedForDeletion set to TRUE) or returns an error message upon failure.

### Parameters

#### **-HardwareProfile<HardwareProfile>**

Specifies a hardware profile object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).



---

## Notes

- Requires a VMM hardware profile object, which can be retrieved by using the Get-SCHardwareProfile cmdlet.

## Examples

### 1: Remove a specific hardware profile from the library.

The first command gets the hardware profile object named NewHWProfile01 from the VMM library and stores the object in the \$HWProfile variable.

The second command deletes NewHWProfile01 from the library, prompting the user for confirmation before completing the operation.

```
PS C:\> $HWProfile = Get-SCHardwareProfile | where { $_.Name -eq "NewHWProfile01"}  
PS C:\> Remove-SCHardwareProfile -HardwareProfile $HWProfile -Confirm
```

### 2: Remove all hardware profiles without being prompted to confirm each deletion.

This command gets all hardware profile objects in the library and passes each profile object to the Remove-SCHardwareProfile cmdlet, which removes each hardware profile. By not using the Confirm parameter, you will not be prompted to confirm whether you want to delete these hardware profile objects.

```
PS C:\> Get-SCHardwareProfile | Remove-SCHardwareProfile
```

## Related topics

[Get-SCHardwareProfile](#)

[New-SCHardwareProfile](#)

[Set-SCHardwareProfile](#)

---

# Remove-SCISO

---

## Remove-SCISO

Removes an ISO file from the VMM library.

### Syntax

Parameter Set: Default

```
Remove-SCISO [-ISO] <ISO> [-Force] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The Remove-SCISO cmdlet removes an ISO file from the VMM library and deletes the ISO file on the library server.

If the ISO is attached to a virtual machine, template, or hardware profile (and if you do not use the Force parameter), VMM lists the container that includes the ISO and prompts you to confirm that you want to remove the ISO:

- If you reply Yes, VMM removes the association between the ISO and the container to which it is attached, and then deletes the ISO object from VMM.
- If you reply No, the operation is cancelled.

This cmdlet returns the object upon success (with the property MarkedForDeletion set to TRUE) or returns an error message upon failure.

### Parameters

#### **-Force**

Forces the operation to complete.

For example:

```
- Remove-SCSCVMHost -Force
```

Forces the removal of a host object from the VMM database.

```
- Stop-SCVirtualMachine -Force
```

Stops a virtual machine.

---

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-ISO<ISO>**

Specifies an ISO object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

---

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Confirm**

Prompts you for confirmation before executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

---

|                             |       |
|-----------------------------|-------|
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## Notes

- Requires a VMM ISO object, which can be retrieved by using the Get-SCISO cmdlet.

## Examples

### 1: Remove an ISO object and delete the corresponding .iso file.

The first command gets the ISO object named OsISO.iso from LibraryServer01 and stores the ISO object in the \$ISO variable.

The second command removes the ISO object from the library and deletes the corresponding .iso file from the file system on the library server.

```
PS C:\> $ISO = Get-SCISO -VMMServer "VMMServer01.Contoso.com" | where { $_.Name -eq "OsISO.iso" -and $_.LibraryServer.Name -eq "LibraryServer01.Contoso.com" }
PS C:\> Remove-SCISO -ISO $ISO
```

### 2: Remove multiple ISO objects from the library.

The first command gets all ISO objects whose name includes the string "OsISO" and stores these ISO objects in the \$ISOs variable.

The second command passes each ISO object in \$ISOs to the Remove-ISO cmdlet, which removes each ISO object from the library and deletes the corresponding .iso file from the file system on the library server.

```
PS C:\> $ISOs = Get-SCISO -VMMServer "VMMServer01.Contoso.com" | where { $_.Name -match "OsISO" }
PS C:\> $ISOs | Remove-SCISO
```

## Related topics

[Get-SCISO](#)

---

[Set-SCISO](#)

---

# Remove-SCLibraryServer

---

## Remove-SCLibraryServer

Removes a library server from VMM.

### Syntax

Parameter Set: Default

```
Remove-SCLibraryServer [-LibraryServer] <LibraryServer> -Credential <VMMCredential> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-Confirm] [-WhatIf] [<CommonParameters>]
```

### Detailed Description

The Remove-SCLibraryServer cmdlet removes a library server object (and all library objects on that library server) from the Virtual Machine Manager (VMM) database. Library objects that have a corresponding file (such as .vhd or .vmdk files) stored on the server's file system are not removed from the file system by this cmdlet.

This cmdlet operates as follows:

- If this library server is also the VMM server, you cannot remove the library server, so the remove library server operation will fail.
- If this computer is both a library server and a host, this cmdlet removes only the library server feature from VMM, but the computer continues to function as a host.
- If this computer is only a library server (not also a host or a VMM server), the library server is removed from VMM.

This cmdlet returns the object upon success (with the property MarkedForDeletion set to TRUE) or returns an error message upon failure.

### Parameters

#### **-Credential<VMMCredential>**

Specifies a credential object or, for some cmdlets, a Run As account object that contains the user name and password of an account that has permission to perform this action. Or, in the case of Restart-SCJob, has permission to complete a restarted task.

For more information about the PSCredential object, type: "Get-Help Get-Credential".

---

For more information about Run As accounts, type: "Get-Help New-SCRunAsAccount".

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-LibraryServer<LibraryServer>**

Specifies a VMM library server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |



---

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-Confirm**

Prompts you for confirmation before executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## -WhatIf

Describes what would happen if you executed the command without actually executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Examples

### 1: Remove a library server object from VMM.

The first command prompts you for credentials. When the dialog box appears, type the user name and password for either a local Administrator account or a domain account with administrator rights on the library server.

The second command retrieves the library server object named LibraryServer01 on VMMServer01 and stores it in the \$LibServ variable.

The third command removes the library server object, and all library shares on this server, from the VMM library. When the Remove-SCLibraryServer cmdlet is used with the -LibraryServer parameter as shown in this example, you can pass only one library server object to the cmdlet.

```
PS C:\> $Creds = Get-Credential
```

```
PS C:\> $LibServ = Get-SCLibraryServer -VMMServer "VMMServer1.Contoso.com" -ComputerName  
"LibraryServer01.Contoso.com"
```

```
PS C:\> Remove-SCLibraryServer -LibraryServer $LibServ -Credential $Creds
```

### 2: Remove multiple library server objects that have a specific string in their name.

The first command prompts you for credentials. When the dialog box appears, type the user name and password for either a local Administrator account or a domain account with administrator rights on the library server.

The second command gets all library server objects from VMMServer01 with names that include the string "LibraryServer" and stores the returned objects in the \$LibServers variable (an object array).

---

The third command passes each library server object in `$LibServers` to `Remove-SCLibraryServer`, which removes each object from VMM.

```
PS C:\> $Creds = Get-Credential
PS C:\> $LibServers = Get-SCLibraryServer -VMMServer "VMMServer01.Contoso.com" | where {
$_Name -match "LibraryServer" }
PS C:\> $LibServers | Remove-SCLibraryServer -Credential $Creds
```

### 3: Remove a highly available library server and all of its nodes.

The first command uses `Get-Credential` to prompt you to supply a user name and password and stores your credentials in `$Credential`. The required credentials for this operation are either a local Administrator account or a domain account with administrator rights on the library server. The following commands use `$Credential` to pass your credentials to each cmdlet that requires credentials.

The second command uses the `Find-SCCluster` cmdlet to confirm that `HAFileServer01` is a highly available file server and stores the cluster object in the `$Cluster` variable.

The third command removes the highly available file server (by specifying its name) as a library server from VMM. The command uses the `-RunAsynchronously` parameter to return control to the shell immediately (before this command completes) because the last command does not need to wait until after this command finishes.

The last command uses a `foreach` loop to pass each object stored in `$Cluster.ClusterNodes` to the `Remove-LibraryServer` cmdlet, which removes each node from VMM. The command uses the `-RunAsynchronously` parameter to return control to the shell immediately. For more information about the Windows PowerShell `foreach` loop statement, type: "`Get-Help about_ForEach`".

```
PS C:\> $Credential = Get-Credential
PS C:\> $Cluster = Find-SCCluster -ComputerName "HAFileServer01.Contoso.com" -Credential
$Credential
PS C:\> Remove-LibraryServer -LibraryServer "HAFileServer01.Contoso.com" -Credential
$Credential -RunAsynchronously
PS C:\> ForEach ($Node in $Cluster.ClusterNodes) {Remove-LibraryServer -LibraryServer $Node
-Credential $Credential -RunAsynchronously}
```

## Related topics

[Add-SCLibraryServer](#)

[Get-SCLibraryServer](#)

[Remove-SCLibraryShare](#)

[Set-SCLibraryServer](#)

---

# Remove-SCLibraryShare

---

## Remove-SCLibraryShare

Removes a library share from VMM but does not delete the share from the Windows file system.

### Syntax

Parameter Set: Default

```
Remove-SCLibraryShare [-LibraryShare] <LibraryShare> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The Remove-SCLibraryShare removes a library share from the VMM library. This cmdlet does not remove any shares or files from the file system on the computer.

This cmdlet returns the object upon success (with the property MarkedForDeletion set to TRUE) or returns an error message upon failure.

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### **-LibraryShare<LibraryShare>**

Specifies a VMM library share object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Confirm**

Prompts you for confirmation before executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Examples**

### **1: Remove a library share object from the VMM library,**

The first command gets the library share object named AllVHDs on LibraryServer01 from the VMM library on VMMServer01 and then stores the object in the \$LibShare variable.

The second command removes the library share object and all library objects in this share from the VMM library but does not delete the share or its contents from the file system on the library server.

```
PS C:\> $LibShare = Get-SCLibraryShare -VMMServer "VMMServer01.Contoso.com" | where {
$_.LibraryServer.name -eq "LibraryServer01.Contoso.com" -and $_.Name -eq "AllVHDs" }
PS C:\> Remove-SCLibraryShare -LibraryShare $LibShare
```

---

## 2: Remove multiple library share objects from the VMM library.

The first command gets all library share objects on LibraryServer01 whose name includes the string "vhd" from the VMM library on VMMServer01 and then stores these share objects in the \$LibShares variable (an object array).

The second command passes each library share object in \$LibShares to Remove-SCLibraryShare. The cmdlet removes each of the library share objects and all objects in the share from the VMM library but does not delete the corresponding shares or their contents from the file system on the library server.

```
PS C:\> $LibShares = Get-SCLibraryShare -VMMServer "VMMServer01.Contoso.com" | where {
$_ .LibraryServer.Name -eq "LibraryServer01.Contoso.com" -and $_.Name -match "vhd" }
```

```
PS C:\> $LibShares | Remove-SCLibraryShare
```

### Related topics

[Add-SCLibraryShare](#)

[Get-SCLibraryShare](#)

[Read-SCLibraryShare](#)

---

# Remove-SCLoadBalancer

---

## Remove-SCLoadBalancer

Removes a load balancer from VMM.

### Syntax

Parameter Set: Default

```
Remove-SCLoadBalancer [-LoadBalancer] <LoadBalancer> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The Remove-SCLoadBalancer cmdlet removes a load balancer from Virtual Machine Manager (VMM).

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### **-LoadBalancer<LoadBalancer>**

Specifies a load balancer object.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|



|                             |                |
|-----------------------------|----------------|
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

---

|                             |                |
|-----------------------------|----------------|
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-Confirm**

Prompts you for confirmation before executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **LoadBalancer**

---

## Examples

### 1: Remove a specific load balancer.

The first command gets the load balancer object at address LB01.Contoso.com and stores the object in the \$LoadBalancer variable.

The second command removes the load balancer in \$LoadBalancer from VMM.

```
PS C:\> $LoadBalancer = Get-SCLoadBalancer -LoadBalancerAddress "LB01.Contoso.com"
```

```
PS C:\> Remove-SCLoadBalancer -LoadBalancer $LoadBalancer
```

## Related topics

[Add-SCLoadBalancer](#)

[Get-SCLoadBalancer](#)

[Read-SCLoadBalancer](#)

[Set-SCLoadBalancer](#)

[Test-SCLoadBalancer](#)

---

# Remove-SCLoadBalancerTemplate

---

## Remove-SCLoadBalancerTemplate

Removes a load balancer template from a service template.

### Syntax

Parameter Set: Default

```
Remove-SCLoadBalancerTemplate -LoadBalancerTemplate <LoadBalancerTemplate> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

### Detailed Description

The Remove-SCLoadBalancerTemplate cmdlet removes a load balancer template from a service template.

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### **-LoadBalancerTemplate<LoadBalancerTemplate>**

Specifies a load balancer template object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Confirm**

Prompts you for confirmation before executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Examples**

### **1: Remove a load balancer template from a service template.**

The first command gets the service template object named ServiceTemplate01 and stores the object in the \$ServiceTemplate variable.

The second command gets the load balancer template for the service template in \$ServiceTemplate and stores the object in the \$LoadBalancerTemplate variable.

The last command removes the load balancer template stored in \$LoadBalancer from the service template stored in \$ServiceTemplate.

```
PS C:\> $ServiceTemplate = Get-SCServiceTemplate -Name "ServiceTemplate01"
```

```
PS C:\> $LoadBalancerTemplate = Get-SCLoadBalancerTemplate -ServiceTemplate $ServiceTemplate
```

```
PS C:\> Remove-SCLoadBalancerTemplate -LoadBalancerTemplate $LoadBalancerTemplate
```

---

## Related topics

[Get-SCLoadBalancerTemplate](#)

[Get-SCServiceTemplate](#)

[New-SCLoadBalancerTemplate](#)

[Set-SCLoadBalancerTemplate](#)

---

# Remove-SCLoadBalancerVIP

---

## Remove-SCLoadBalancerVIP

Removes a load balancer VIP from a load balancer.

### Syntax

Parameter Set: Default

```
Remove-SCLoadBalancerVIP [-LoadBalancerVIP] <LoadBalancerVIP> [-JobVariable <String> ] [-  
PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [-Confirm] [-WhatIf]  
[ <CommonParameters>]
```

### Detailed Description

The Remove-SCLoadBalancerVIP cmdlet removes a load balancer virtual IP (VIP) from a load balancer.

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### **-LoadBalancerVIP<LoadBalancerVIP>**

Specifies a virtual IP (VIP) in a load balancer.



|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

---

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-Confirm**

Prompts you for confirmation before executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- 
- **LoadBalancerVIP**

## Examples

### 1: Remove a virtual IP (VIP) from a load balancer.

The first command gets the load balancer object with the address LB01.Contoso.com and stores the object in the \$LoadBalancer variable.

The second command gets the load balancer VIP for the load balancer stored in \$LoadBalancer with the IP address of 10.0.0.1.

The last command removes the load balancer VIP from the load balancer.

```
PS C:\> $LoadBalancer = Get-SCLoadBalancer -LoadBalancerAddress "LB01.contoso.com"
```

```
PS C:\> $VIP = Get-SCLoadBalancerVIP -LoadBalancer $LoadBalancer -IPAddress "10.0.0.1"
```

```
PS C:\> Remove-SCLoadBalancerVIP -LoadBalancerVIP $VIP
```

## Related topics

[Get-SCLoadBalancerVIP](#)

[New-SCLoadBalancerVIP](#)

[Read-SCLoadBalancerVIP](#)

---

# Remove-SCLoadBalancerVIPMember

---

## Remove-SCLoadBalancerVIPMember

Removes a member from a load balancer VIP.

### Syntax

Parameter Set: Default

```
Remove-SCLoadBalancerVIPMember [-LoadBalancerVIPMember] <LoadBalancerVIPMember> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The Remove-SCLoadBalancerVIPMember removes a member from a load balancer virtual IP (VIP).

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### **-LoadBalancerVIPMember<LoadBalancerVIPMember>**

Specifies a member of a virtual IP (VIP) in a load balancer.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

|                             |                |
|-----------------------------|----------------|
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

|                             |                |
|-----------------------------|----------------|
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-Confirm**

Prompts you for confirmation before executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **LoadBalancerVIPMember**

---

## Examples

### 1: Remove a member from a load balancer virtual IP (VIP).

The first command gets the load balancer object with the address of LB01.Contoso.com and stores the object in the \$LoadBalancer variable.

The second command gets the load balancer VIP with the IP address 10.0.0.1 for the load balancer stored in \$LoadBalancer and stores the object in the \$VIP variable.

The third command gets the VIP member for the load balancer VIP stored in \$VIP with the address of 10.0.0.1 and stores the object in the \$VIPMember variable.

The last command removes the VIP member stored in \$VIPMember from the load balancer VIP.

```
PS C:\> $LoadBalancer = Get-SCLoadBalancer -LoadBalancerAddress "LB01.Contoso.com"
PS C:\> $VIP = Get-SCLoadBalancerVIP -LoadBalancer $LoadBalancer -IPAddress "10.0.0.1"
PS C:\> $VIPMember = Get-SCLoadBalancerVIPMember -LoadBalancerVIP $VIP -IPAddress "10.0.0.1"
PS C:\> Remove-SCLoadBalancerVIPMember -LoadBalancerVIPMember $VIPMember
```

## Related topics

[Disable-SCLoadBalancerVIPMember](#)

[Enable-SCLoadBalancerVIPMember](#)

[Get-SCLoadBalancerVIP](#)

[Get-SCLoadBalancerVIPMember](#)

[New-SCLoadBalancerVIPMember](#)

---

# Remove-SCLoadBalancerVIPTemplate

---

## Remove-SCLoadBalancerVIPTemplate

Deletes a load balancer VIP template from VMM.

### Syntax

Parameter Set: Default

```
Remove-SCLoadBalancerVIPTemplate [-LoadBalancerVIPTemplate] <LoadBalancerVIPTemplate> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The Remove-SCLoadBalancerVIPTemplate deletes a load balancer virtual IP (VIP) template from Virtual Machine Manager (VMM).

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### **-LoadBalancerVIPTemplate<LoadBalancerVIPTemplate>**

Specifies a load balancer virtual IP (VIP) template.



---

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

---

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-Confirm**

Prompts you for confirmation before executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- 
- **LoadBalancerVIPTemplate**

## Examples

### 1: Delete a load balancer virtual IP (VIP) template.

The first command gets the load balancer VIP template named VIPtemplate01 and stores the object in the \$VIPTemplate variable.

The second command removes the load balancer VIP template stored in \$VIPTemplate.

```
PS C:\> $VIPTemplate = Get-SCLoadBalancerVIPTemplate -Name "VIPTemplate01"
```

```
PS C:\> Remove-SCLoadBalancerVIPTemplate -LoadBalancerVIPTemplate $VIPTemplate
```

## Related topics

[Get-SCLoadBalancerVIPTemplate](#)

[New-SCLoadBalancerVIPTemplate](#)

[Set-SCLoadBalancerVIPTemplate](#)

---

# Remove-SCLogicalNetwork

---

## Remove-SCLogicalNetwork

Deletes a logical network object.

### Syntax

Parameter Set: Default

```
Remove-SCLogicalNetwork [-LogicalNetwork] <LogicalNetwork> [-JobVariable <String> ] [-  
PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [-Confirm] [-WhatIf]  
[ <CommonParameters>]
```

### Detailed Description

The Remove-SCLogicalNetwork cmdlet deletes a Virtual Machine Manager (VMM) logical network object.

NOTE: If any logical network definition, virtual machine template, or service template object references a specific logical network object, that logical network object will not be deleted.

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-LogicalNetwork<LogicalNetwork>**

Specifies a logical network. A logical network is a named grouping of IP subnets and VLANs that is used to organize and simplify network assignments.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-Confirm**

Prompts you for confirmation before executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **LogicalNetwork**

## Notes

- Requires a VMM logical network object, which can be retrieved by using the Get-SCLogicalNetwork cmdlet.

## Examples

### 1: Delete a logical network

The first command gets the logical network object named LogicalNetwork01 and stores it in the \$LogicalNetwork variable.

The second command deletes the logical network object stored in \$LogicalNetwork.

```
PS C:\> $LogicalNetwork = Get-SCLogicalNetwork "LogicalNetwork01"
```

```
PS C:\> Remove-SCLogicalNetwork -LogicalNetwork $LogicalNetwork
```

## Related topics

[Get-SCLogicalNetwork](#)

[New-SCLogicalNetwork](#)

[Set-SCLogicalNetwork](#)

---

# Remove-SCLogicalNetworkDefinition

---

## Remove-SCLogicalNetworkDefinition

Deletes a logical network definition.

### Syntax

Parameter Set: Default

```
Remove-SCLogicalNetworkDefinition [-LogicalNetworkDefinition] <LogicalNetworkDefinition> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The Remove-SCLogicalNetworkDefinition cmdlet deletes a logical network definition (also called a network site).

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### **-LogicalNetworkDefinition<LogicalNetworkDefinition>**

Specifies a logical network definition (also called a network site) that contains the subnet that the IP address pool serves as specified by the Subnet parameter.



|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-Confirm**

Prompts you for confirmation before executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- 
- **LogicalNetworkDefiniton**

## Notes

- Requires a VMM logical network definition object, which can be retrieved by using the Get-SCLogicalNetworkDefinition cmdlet.

## Examples

### 1: Delete a logical network definition.

The first command gets the logical network definition named "Logical Network Definition 01" and stores it in the \$Definition variable.

The second command deletes the logical network definition stored in the \$Definition variable.

```
PS C:\> $Definition = Get-SCLogicalNetworkDefinition -Name "Logical Network Definition 01"
```

```
PS C:\> Remove-SCLogicalNetworkDefinition -LogicalNetworkDefinition $Definition
```

## Related topics

[Get-SCLogicalNetworkDefinition](#)

[New-SCLogicalNetworkDefinition](#)

[Set-SCLogicalNetworkDefinition](#)

---

# Remove-SCLogicalSwitch

---

## Remove-SCLogicalSwitch

Removes a logical switch.

### Syntax

Parameter Set: Default

```
Remove-SCLogicalSwitch [-LogicalSwitch] <LogicalSwitch> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The Remove-LogicalSwitch cmdlet deletes a logical switch.

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### **-LogicalSwitch<LogicalSwitch>**

Specifies a logical switch object.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

|                             |                |
|-----------------------------|----------------|
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

---

|                             |                |
|-----------------------------|----------------|
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-Confirm**

Prompts you for confirmation before executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

---

## Examples

### 1: Remove a logical switch.

This command gets the logical switch object named LogicalSwitch01 and then uses the pipeline operator to pass the object to Remove-SCLogicalSwitch which deletes the logical switch object.

```
PS C:\> $LogicalSwitch = Get-SCLogicalSwitch -Name "LogicalSwitch01" | Remove-SCLogicalSwitch
```

### Related topics

[Get-SCLogicalSwitch](#)

[New-SCLogicalSwitch](#)

[Set-SCLogicalSwitch](#)

---

# Remove-SCMACAddressPool

---

## Remove-SCMACAddressPool

Deletes a MAC address pool.

### Syntax

Parameter Set: Default

```
Remove-SCMACAddressPool [-MACAddressPool] <MACAddressPool> [-JobVariable <String> ] [-  
PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [-Confirm] [-WhatIf]  
[ <CommonParameters>]
```

### Detailed Description

The Remove-SCMACAddressPool cmdlet deletes a Virtual Machine Manager (VMM) MAC address pool.

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### **-MACAddressPool<MACAddressPool>**

Specifies a MAC address pool.



|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-Confirm**

Prompts you for confirmation before executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- 
- **MACAddressPool**

## Notes

- Requires a VMM MAC address pool object, which can be retrieved by using the Get-SCMACAddressPool cmdlet.

## Examples

### 1: Delete a MAC address pool.

The first command gets the host group named "All Hosts\HostGroup02\Production" and stores it in the \$HostGroup variable.

The second command gets the MAC address pool named "MAC Address Pool 01" associated with the host group stored in the \$HostGroup variable (including its parent host group if inheritance is enabled) and stores it in the \$MACPool variable.

The third command deletes the MAC address pool stored in the \$MACPool variable.

```
PS C:\> $HostGroup = Get-SCVMHostGroup | where { $_.Path -eq "All  
Hosts\HostGroup02\Production" }
```

```
PS C:\> $MACPool = Get-SCMACAddressPool -VMHostGroup $HostGroup -Name "MAC Address Pool 01"
```

```
PS C:\> Remove-SCMACAddressPool -MACAddressPool $MACPool
```

## Related topics

[Get-SCMACAddressPool](#)

[New-SCMACAddressPool](#)

[Set-SCMACAddressPool](#)

---

# Remove-SCNativeUplinkPortProfile

---

## Remove-SCNativeUplinkPortProfile

Removes a native uplink port profile.

### Syntax

Parameter Set: Default

```
Remove-SCNativeUplinkPortProfile [-NativeUplinkPortProfile] <NativeUplinkPortProfile> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The Remove-SCNativeUplinkPortProfile cmdlet deletes a native uplink port profile.

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### **-NativeUplinkPortProfile<NativeUplinkPortProfile>**

Specifies a native uplink port profile object.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

|                             |                |
|-----------------------------|----------------|
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

---

|                             |                |
|-----------------------------|----------------|
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-Confirm**

Prompts you for confirmation before executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

---

## Examples

### 1: Remove a native uplink port profile.

This command gets the native uplink port profile object named NativeUplinkPortProfile01 and uses the pipeline operator to pass the object to Remove-SCNativeUplinkPortProfile which deletes the native uplink port profile.

```
PS C:\> Get-SCNativeUplinkPortProfile -Name "NativeUplinkPortProfile01" | Remove-SCNativeUplinkPortProfile
```

## Related topics

[Get-SCNativeUplinkPortProfile](#)

[New-SCNativeUplinkPortProfile](#)

[Set-SCNativeUplinkPortProfile](#)

---

# Remove-SCNetworkGateway

---

## Remove-SCNetworkGateway

Removes a network gateway object from VMM.

### Syntax

Parameter Set: Default

```
Remove-SCNetworkGateway [-NetworkGateway] <NetworkGateway> [-JobVariable <String> ] [-  
PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [-Confirm] [-WhatIf]  
[ <CommonParameters>]
```

### Detailed Description

The Remove-SCNetworkGateway cmdlet deletes a network gateway object from Virtual Machine Manager (VMM).

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### **-NetworkGateway<NetworkGateway>**

Specifies a network gateway object.

To obtain a network gateway object, use the Get-SCNetworkGateway cmdlet.



|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-Confirm**

Prompts you for confirmation before executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Related topics**

[Add-SCNetworkGateway](#)

---

[Get-SCNetworkGateway](#)

[Read-SCNetworkGateway](#)

[Set-SCNetworkGateway](#)

[Test-SCNetworkGateway](#)

---

# Remove-SCOperatingSystem

---

## Remove-SCOperatingSystem

Deletes an operating system from an application profile.

### Syntax

Parameter Set: Default

```
Remove-SCOperatingSystem -ApplicationProfile <ApplicationProfile> -OperatingSystem  
<OperatingSystem> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-  
VMMServer <ServerConnection> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The Remove-SCOperatingSystem cmdlet deletes an operating system from an application profile.

### Parameters

#### **-ApplicationProfile<ApplicationProfile>**

Specifies an application profile object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-OperatingSystem<OperatingSystem>**

Specifies the type of operating system for a virtual machine. To list the names of all available operating systems in VMM, type: "Get-SCOperatingSystem".

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-Confirm**

Prompts you for confirmation before executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

---

|                             |       |
|-----------------------------|-------|
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## Examples

### 1: Remove all operating systems from a specified application profile

The first command gets the application profile object named SvcWebAppProfile01 and stores it in the \$AppProfile variable.

The second command gets the first operating system object stored in the OperatingSystems property of the application profile stored in \$AppProfile and stores the object in the \$OS variable.

The last command removes the operating system stored in \$OS from the application profile stored in \$AppProfile.

```
PS C:\> $AppProfile = Get-SCApplicationProfile -Name "SvcWebAppProfile01"
```

```
PS C:\> $OS = $AppProfile.OperatingSystems[0]
```

```
PS C:\> Remove-SCOperatingSystem -ApplicationProfile $AppProfile -OperatingSystem $OS
```

## Related topics

[Add-SCOperatingSystem](#)

---

# Remove-SCOpsMgrConnection

---

## Remove-SCOpsMgrConnection

Removes the Operations Manager connection from VMM.

### Syntax

Parameter Set: Default

```
Remove-SCOpsMgrConnection [-Force] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

### Detailed Description

The Remove-SCOpsMgrConnection cmdlet removes the existing Operations Manager connection from Virtual Machine Manager (VMM).

### Parameters

#### -Force

Forces the operation to complete.

For example:

- Remove-SCSCVMHost -Force

Forces the removal of a host object from the VMM database.

- Stop-SCVirtualMachine -Force

Stops a virtual machine.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |



---

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-Confirm**

Prompts you for confirmation before executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Examples

### 1: Remove the Operations Manager connection from VMM.

This command removes the Operations Manager connection from VMM.

```
PS C:\> Remove-SCOpsMgrConnection
```

### 2: Remove an Operations Manager connection that is not accessible from VMM.

This command removes an Operations Manager connection from VMM when the Operations Manager server is unavailable or no longer accessible from VMM.

NOTE: Removing the connection to an Operations Manager server that is unavailable or no longer accessible from VMM will not remove VMM-related artifacts, such as internal connectors, from Operations Manager.

```
PS C:\> Remove-SCOpsMgrConnection -Force
```

## Related topics

[Get-SCOpsMgrConnection](#)

[New-SCOpsMgrConnection](#)

[Set-SCOpsMgrConnection](#)

[Write-SCOpsMgrConnection](#)

---

# Remove-SCPortClassification

---

## Remove-SCPortClassification

Removes a port classification.

### Syntax

Parameter Set: Default

```
Remove-SCPortClassification [-PortClassification] <PortClassification> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The Remove-SCPortClassification cmdlet deletes a port classification.

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### **-PortClassification<PortClassification>**

Specifies the port classification of the switch to which the virtual network adapter connects.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

|                             |                |
|-----------------------------|----------------|
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

|                             |                |
|-----------------------------|----------------|
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-Confirm**

Prompts you for confirmation before executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

---

## Examples

### 1: Remove a port classification.

This command gets the port classification object named PortClass01 and uses the pipeline operator to pass the object to Remove-SCPortClassification which deletes the port classification object.

```
PS C:\> Get-SCPortClassification -Name "PortClass01" | Remove-SCPortClassification
```

## Related topics

[Get-SCPortClassification](#)

[New-SCPortClassification](#)

[Set-SCPortClassification](#)

---

# Remove-SCPXEserver

---

## Remove-SCPXEserver

Removes a PXEserver object from the VMM database.

### Syntax

Parameter Set: Default

```
Remove-SCPXEserver [-PXEserver] <PxeServer> -Credential <VMMCredential> [-Force] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Remove-SCPXEserver cmdlet removes a PXEserver object from the Virtual Machine Manager (VMM) database, and uninstalls the VMM agent from the Windows Deployment Services computer. This cmdlet does not remove the Windows Deployment Services role from the computer.

### Parameters

#### **-Credential<VMMCredential>**

Specifies a credential object or, for some cmdlets, a Run As account object that contains the user name and password of an account that has permission to perform this action. Or, in the case of Restart-SCJob, has permission to complete a restarted task.

For more information about the PSCredential object, type: "Get-Help Get-Credential".

For more information about Run As accounts, type: "Get-Help New-SCRunAsAccount".

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |



---

## **-Force**

Forces the operation to complete.

For example:

- Remove-SCSCVMHost -Force

Forces the removal of a host object from the VMM database.

- Stop-SCVirtualMachine -Force

Stops a virtual machine.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | false |
| Position? | named |

|                             |       |
|-----------------------------|-------|
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PXEServer<PxeServer>**

Specifies a PXE server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | false |
| Position? | named |

---

|                             |                |
|-----------------------------|----------------|
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Examples

### 1: Remove a VMM PXE server.

The first command uses the Get-Credential cmdlet to prompt you to supply a user name and password and stores the provided credentials in the \$Credential variable. The account must have local Administrator permissions on the PXE server.

The second command gets the PXE server object named WDSSEServer01 and stores the object in the \$PXEServer variable.

The third command removes the PXE server object stored in \$PXEServer. As this command is processed, \$Credential provides credentials to Remove-SCPXESEServer, and the Confirm parameter prompts you to confirm that you do want to remove this PXE server from VMM.

```
PS C:\> $Credential = Get-SCRunAsAccount "RunAsAcct01"
```

```
PS C:\> $PXEServer = Get-SCPXESEServer -ComputerName "WDSSEServer01.Contoso.com"
```

```
PS C:\> Remove-SCPXESEServer -PXEServer $PXEServer -Credential $Credential -Confirm
```

## Related topics

[Add-SCPXESEServer](#)

[Get-SCPXESEServer](#)

---

# Remove-SCRunAsAccount

---

## Remove-SCRunAsAccount

Removes a VMM Run As account.

### Syntax

Parameter Set: Default

```
Remove-SCRunAsAccount [-RunAsAccount] <RunAsAccount> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The Remove-SCRunAsAccount cmdlet removes a Run As account from Virtual Machine Manager (VMM).

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsAccount<RunAsAccount>**

Specifies a Run As account that contains credentials with permission to perform this action.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-Confirm**

Prompts you for confirmation before executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

---

## Examples

### 1. Remove a Run As account

The first command gets the Run As account object named RunAsAccount01 and stores the object in the \$RunAsAccount variable.

The second command removes the Run As account stored in \$RunAsAccount.

```
PS C:\> $RunAsAccount = Get-SCRunAsAccount -Name "RunAsAccount01"
```

```
PS C:\> Remove-SCRunAsAccount -RunAsAccount $RunAsAccount
```

### 2: Remove a Run As account using the pipeline operator.

This command gets the Run As account object named RunAsAccount02 and uses the pipeline operator to pass the object to the Remove-SCRunAsAccount cmdlet which removes the account.

```
PS C:\> Get-SCRunAsAccount -Name "RunAsAccount02" | Remove-SCRunAsAccount
```

## Related topics

[Get-SCRunAsAccount](#)

[New-SCRunAsAccount](#)

[Disable-SCRunAsAccount](#)

---

# Remove-SCScript

---

## Remove-SCScript

Removes a script object from VMM.

### Syntax

Parameter Set: Default

```
Remove-SCScript [-Script] <Script> [-Force] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The Remove-SCScript cmdlet removes one or more script objects from the VMM library and deletes the corresponding script file on the library server.

If the script is attached to a template or hardware profile (and if you do not use the Force parameter), Virtual Machine Manager (VMM) lists the container that contains the script and prompts you to confirm that you want to remove the script:

- If you reply Yes, VMM removes the association between the script and the container to which it is attached, and then deletes the script object from VMM.
- If you reply No, the operation is cancelled.

This cmdlet returns the object upon success (with the property MarkedForDeletion set to TRUE) or returns an error message upon failure.

### Parameters

#### **-Force**

Forces the operation to complete.

For example:

- Remove-SCSCVMHost -Force

Forces the removal of a host object from the VMM database.

- Stop-SCVirtualMachine -Force

Stops a virtual machine.



|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Script<Script>**

Specifies a VMM script object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-Confirm**

Prompts you for confirmation before executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

---

|                             |       |
|-----------------------------|-------|
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## Examples

### 1: Remove a script object and delete the corresponding script file.

The first command gets the script object named AddHost.ps1 from the VMM library on VMMServer01 and stores the object in the array named \$Scripts. More than one file with the same name might exist if more than one container for scripts exists on the specified library server.

The second command counts the number of scripts in \$Scripts and displays the results to the user.

The third command passes each script object in \$Scripts to the Select-Object cmdlet, which selects the name and share path for each script in the array. The command then passes these results to the Format-List cmdlet to display each script name, and its share path, to the user.

The last command deletes the first object in the \$Scripts array and uses the Force parameter to ensure that the script object is removed from the VMM database and the corresponding script file is deleted from the file system on the library server.

```
PS C:\> $Scripts = @(Get-SCScript -VMMServer "VMMServer01.Contoso.com" | where {
$_.LibraryServer.Name -eq "LibraryServer01.Contoso.com" -and $_.Name -eq "AddHost.ps1"} )
PS C:\> $Scripts.Count
PS C:\> $Scripts | select Name,SharePath | Format-List
PS C:\> Remove-SCScript -Script $Scripts[0] -Force
```

### 2: Remove multiple scripts from the library.

The first command gets all script objects whose names include the string "Sysprep" from VMMServer01 and then stores these objects in the array named \$Scripts.

The second command passes each script object in \$Scripts to Remove-Script, which removes each script object from the library and deletes each corresponding script file from the file system on the library server.

The Confirm parameter prompts you to confirm that you do want to remove these scripts. You have the option to confirm the deletion of all scripts at once or to confirm the deletion of each script one-by-one.

---

```
PS C:\> $Scripts = Get-SCScript -VMMServer "VMMServer01.Contoso.com" | where { $_.Name -
match "Sysprep" }
PS C:\> $Scripts | Remove-SCScript -Confirm
```

## Related topics

[Get-SCScript](#)

[Set-SCScript](#)

---

# Remove-SCScriptCommand

---

## Remove-SCScriptCommand

Deletes a script command from an application profile, application deployment, or host profile.

### Syntax

Parameter Set: Default

```
Remove-SCScriptCommand -ScriptCommand <SCScriptCommand> [-Force] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The Remove-SCScriptCommand deletes a script command from an application profile, application deployment, or host profile.

### Parameters

#### -Force

Forces the operation to complete.

For example:

- Remove-SCSCVMHost -Force

Forces the removal of a host object from the VMM database.

- Stop-SCVirtualMachine -Force

Stops a virtual machine.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-ScriptCommand<SCScriptCommand>**

Specifies a script command object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-Confirm**

Prompts you for confirmation before executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Examples

### 1: Remove all script commands associated with a specific application profile.

The first command gets the application profile object named SvcWebAppProfile01 and stores the object in the \$AppProfile variable.

The second command gets all script command objects for the application profile stored in \$AppProfile and stores the objects in the \$ScriptCommand array.

The last command uses the pipeline operator to pass each application profile stored in the \$ScriptCommand array to Remove-SCScriptCommand which removes the script command object from application profile SvcWebAppProfile01.

```
PS C:\> $AppProfile = Get-SCApplicationProfile -Name "SvcWebAppProfile01"
```

```
PS C:\> $ScriptCommand = Get-SCScriptCommand -ApplicationProfile $AppProfile
```

```
PS C:\> $ScriptCommand | Remove-SCScriptCommand
```

## Related topics

[Add-SCScriptCommand](#)

[Get-SCScriptCommand](#)

[Set-SCScriptCommand](#)



---

# Remove-SCServerFeature

---

## Remove-SCServerFeature

Removes an operating system role or feature from a guest OS profile.

### Syntax

Parameter Set: OSProfile

```
Remove-SCServerFeature -GuestOSProfile <GuestOSProfile> -ServerFeature <ServerFeature> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-Confirm] [-WhatIf] [<CommonParameters>]
```

Parameter Set: Template

```
Remove-SCServerFeature -ServerFeature <ServerFeature> -VMTemplate <Template> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-Confirm] [-WhatIf] [<CommonParameters>]
```

### Detailed Description

The Remove-SCServerFeature deletes an operating system role or feature from a guest OS profile.

### Parameters

#### **-GuestOSProfile<GuestOSProfile>**

Specifies a guest operating system profile object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-ServerFeature<ServerFeature>**

Specifies a server feature object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VMTemplate<Template>**

Specifies a VMM template object used to create virtual machines.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-Confirm**

Prompts you for confirmation before executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## -WhatIf

Describes what would happen if you executed the command without actually executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Examples

### 1: Remove a server feature from a guest OS profile.

The first command gets the guest OS profile named NewOSProfile01 and stores the object in the \$OSProfile variable.

The second command gets the server feature object named Failover-Clustering and stores the object in the \$Feature variable.

The last command removes the server feature stored in \$Feature from the guest OS profile stored in \$OSProfile.

```
PS C:\> $OSProfile = Get-SCGuestOSProfile -Name "NewOSProfile01"
```

```
PS C:\> $Feature = Get-SCServerFeature -Name "Failover-Clustering"
```

```
PS C:\> Remove-SCServerFeature -GuestOSProfile $OSProfile -ServerFeature $Feature
```

## Related topics

[Add-SCServerFeature](#)

[Get-SCGuestOSProfile](#)

[Get-SCServerFeature](#)

---

# Remove-SCService

---

## Remove-SCService

Deletes a VMM service and all associated virtual machines.

### Syntax

Parameter Set: Default

```
Remove-SCService [-Service] <Service> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Remove-SCService cmdlet deletes a Virtual Machine Manager (VMM) service and all associated virtual machines from the host on which it is deployed. The service must be in a stopped state prior to deleting it. To stop a service, use the Stop-SCService cmdlet.

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Service<Service>**

Specifies a VMM service object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

---

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## Examples

### 1: Remove a specific service deployed on a host.

The first command gets the service object named Service01 on VMMServer01 and stores the object in the \$Service variable.

The second command stops the service stored in \$Service.

The last command removes the service stored in \$Service and deletes the corresponding virtual machine files from the file system. A confirmation prompt is displayed before the service is removed.

```
PS C:\> $Service = Get-SCService -VMMServer "VMMServer01.Contoso.com" -Name "Service01"
PS C:\> Stop-SCService -Service $Service
PS C:\> Remove-SCService -Service $Service -Confirm
```

### 2: Remove all services with names that include a specific string.

The first command gets all service objects that include the string "Service" in their name, and then stores the objects in the \$Services variable.

The second command stops all services stored in \$Service.

The third command removes all service objects contained in \$Services and deletes the corresponding virtual machine files from the file system. A confirmation prompt is displayed before the the service is removed.

```
PS C:\> $Services = Get-SCService -VMMServer "VMMServer01.Contoso.com" | where { $_.Name - Match "Service" }
PS C:\> $Services | Stop-SCService
PS C:\> $Services | Remove-SCService -Confirm
```

---

## Related topics

[Get-SCService](#)

[New-SCService](#)

[Read-SCService](#)

[Resume-SCService](#)

[Set-SCService](#)

[Start-SCService](#)

[Stop-SCService](#)

[Suspend-SCService](#)

[Update-SCService](#)



---

# Remove-SCServiceConfiguration

---

## Remove-SCServiceConfiguration

Deletes a service configuration object from the VMM library.

### Syntax

Parameter Set: Default

```
Remove-SCServiceConfiguration [-ServiceConfiguration] <ServiceConfiguration> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The Remove-SCServiceConfiguration deletes one or more service configuration objects from the VMM library.

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-ServiceConfiguration<ServiceConfiguration>**

Specifies a service configuration object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-Confirm**

Prompts you for confirmation before executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

---

## Examples

### 1: Remove a specific service configuration object from the library.

The first command gets the service configuration object named Service01 and stores the object in the \$SvcConfig variable.

The second command removes the service configuration object stored in \$SvcConfig from the VMM database and deletes the corresponding service configuration object and all other associated objects in the library. A confirmation prompt is displayed before the the service configuration object is removed.

```
PS C:\> $SvcConfig = Get-SCServiceConfiguration -VMMServer "VMMServer01.Contoso.com" -Name "Service01"
```

```
PS C:\> Remove-SCServiceConfiguration -ServiceConfiguration $SvcConfig -Confirm
```

### 2. Remove all service configuration objects

The first command gets all service configuration objects on VMMServer01 and stores the objects in the \$SvcConfigs variable.

The second command removes all the service configuration objects stored in \$SvcConfigs and deletes all other associated objects in the library. A confirmation prompt is displayed before the the service configuration objects are removed.

```
PS C:\> $SvcConfigs = Get-SCServiceConfiguration -VMMServer "VMMServer01.Contoso.com"
```

```
PS C:\> $SvcConfigs | Remove-SCServiceConfiguration -Confirm
```

## Related topics

[Get-SCServiceConfiguration](#)

[New-SCServiceConfiguration](#)

[Set-SCServiceConfiguration](#)

[Update-SCServiceConfiguration](#)

---

# Remove-SCServiceTemplate

---

## Remove-SCServiceTemplate

Deletes a service template from the VMM library.

### Syntax

Parameter Set: Default

```
Remove-SCServiceTemplate [-ServiceTemplate] <ServiceTemplate> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The Remove-SCServiceTemplate cmdlet deletes one or more service templates from the VMM library.

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-ServiceTemplate<ServiceTemplate>**

Specifies a service template object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-Confirm**

Prompts you for confirmation before executing the command.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                             |       |
|-----------------------------|-------|
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## -WhatIf

Describes what would happen if you executed the command without actually executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Examples

### 1: Remove a specific service template from the library.

The first command gets the service template object named ServiceTemplate01 and stores the object in the \$SvcTemplate variable.

The second command removes the service template object in \$SvcTemplate from the VMM database and deletes the corresponding service template object and all other associated objects in the library. A confirmation prompt is displayed before the the service template object is removed.

```
PS C:\> $SvcTemplate = Get-SCServiceTemplate -VMMServer "VMMServer01.Contoso.com" -Name "ServiceTemplate01"
```

```
PS C:\> Remove-SCServiceTemplate -ServiceTemplate $SvcTemplate -Confirm
```

### 2: Remove all service templates from the library.

The first command gets all service template objects on VMMServer01 and stores the objects in the \$SvcTemplates variable.

---

The second command removes all service template objects from the VMM database and deletes the corresponding service template object and all other associated objects in the library. A confirmation prompt is displayed before the service template objects are removed.

```
PS C:\> $SvcTemplates = Get-SCServiceTemplate -VMMServer "VMMServer01.Contoso.com"
```

```
PS C:\> $SvcTemplates | Remove-SCServiceTemplate -Confirm
```

## Related topics

[Get-SCServiceTemplate](#)

[New-SCServiceTemplate](#)

[Read-SCServiceTemplate](#)

[Resolve-SCServiceTemplate](#)

[Set-SCServiceTemplate](#)

[Test-SCServiceTemplate](#)



---

# Remove-SCServicingWindow

---

## Remove-SCServicingWindow

Removes a servicing window definition from VMM.

### Syntax

Parameter Set: Default

```
Remove-SCServicingWindow [-ServicingWindow] <ServicingWindow> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The Remove-SCServicingWindow cmdlet removes a servicing window definition from Virtual Machine Manager (VMM). If the servicing window is assigned to a virtual machine, a host, or a service, the association is also removed.

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-ServicingWindow<ServicingWindow>**

Specifies a servicing window object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-Confirm**

Prompts you for confirmation before executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Notes**

- Requires a VMM servicing window object, which can be retrieved by using the Get-SCServicingWindow cmdlet.

## **Examples**

### **1. Remove a servicing window.**

The first command gets the servicing window object named Backup Staging A and stores the object in the \$SvcWindow variable.

The second command removes the servicing window stored in \$SvcWindow (Backup Staging A).

**IMPORTANT:** All objects subscribing to this service window will lose their subscription.

```
PS C:\> $SvcWindow = Get-SCServicingWindow -Name "Backup Staging A"
```

```
PS C:\> Remove-SCServicingWindow -ServicingWindow $SvcWindow -Confirm
```

---

## Related topics

[Get-SCServicingWindow](#)

[New-SCServicingWindow](#)

[Set-SCServicingWindow](#)

---

# Remove-SCServicingWindowSubscription

---

## Remove-SCServicingWindowSubscription

Removes a servicing window from a virtual machine, host, or service.

### Syntax

Parameter Set: Host

```
Remove-SCServicingWindowSubscription [-ServicingWindow] <ServicingWindow> -VMHost <Host> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-Confirm] [-WhatIf] [<CommonParameters>]
```

Parameter Set: Service

```
Remove-SCServicingWindowSubscription [-ServicingWindow] <ServicingWindow> -Service <Service> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-Confirm] [-WhatIf] [<CommonParameters>]
```

Parameter Set: VM

```
Remove-SCServicingWindowSubscription [-ServicingWindow] <ServicingWindow> -VM <VM> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-Confirm] [-WhatIf] [<CommonParameters>]
```

### Detailed Description

The Remove-SCServicingWindowSubscription cmdlet removes a servicing window from a virtual machine, host, or service.

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                        |       |
|------------------------|-------|
| Aliases                | none  |
| Required?              | false |
| Position?              | named |
| Default Value          | none  |
| Accept Pipeline Input? | false |

|                             |       |
|-----------------------------|-------|
| Accept Wildcard Characters? | false |
|-----------------------------|-------|

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Service<Service>**

Specifies a VMM service object.

|                        |                |
|------------------------|----------------|
| Aliases                | none           |
| Required?              | true           |
| Position?              | named          |
| Default Value          | none           |
| Accept Pipeline Input? | True (ByValue) |

|                             |       |
|-----------------------------|-------|
| Accept Wildcard Characters? | false |
|-----------------------------|-------|

## **-ServicingWindow<ServicingWindow>**

Specifies a servicing window object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | 1     |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VM<VM>**

Specifies a virtual machine object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-VMHost<Host>**

Specifies a virtual machine host object. VMM supports Hyper-V hosts, VMware ESX hosts, and Citrix XenServer hosts.

For more information about each type of host, type: "Get-Help Add-SCVMHost -detailed". See the examples for a specific cmdlet to determine how that cmdlet uses this parameter.

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | true  |
| Position? | named |

|                             |                |
|-----------------------------|----------------|
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-Confirm**

Prompts you for confirmation before executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Notes**

- Requires a VMM servicing window subscription object, which can be retrieved using the Get-SCServicingWindow cmdlet.



---

## Examples

### 1: Remove a servicing window subscription from all virtual machines owned by a specified user.

The first command gets the servicing window object named Backup Staging A and stores the object in the \$SvcWindow variable.

The second command gets all virtual machine objects, selects only the virtual machines that are owned by Contoso\ReneeLo and then uses the pipeline operator to pass the virtual machines to the Remove-SCServicingWindowSubscription cmdlet. Remove-SCServicingWindowSubscription removes all subscriptions for the servicing window stored in \$SvcWindow from each virtual machine that is passed to it.

```
PS C:\> $SvcWindow = Get-SCServicingWindow -Name "Backup Staging A"
```

```
PS C:\> Get-SCVirtualMachine | where {$_.Owner -eq "Contoso\ReneeLo"} | Remove-SCServicingWindowSubscription -ServicingWindow $SvcWindow
```

## Related topics

[Add-SCServicingWindowSubscription](#)

[Get-SCServicingWindowSubscription](#)

---

# Remove-SCSQLDeployment

---

## Remove-SCSQLDeployment

Removes a SQL Server deployment from a SQL profile.

### Syntax

Parameter Set: Default

```
Remove-SCSQLDeployment [-SQLDeployment] <SQLDeployment> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

### Detailed Description

The Remove-SCSQLDeployment cmdlet removes a SQL Server deployment from a SQL profile.

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-SQLDeployment<SQLDeployment>**

Specifies a SQL Server deployment object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

---

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **SQLDeployment**

## Examples

### 1: Remove a SQL Server deployment from a SQL profile.

The first command gets the SQL Server profile object named SQLProfile01 and stores the object in the \$SQLProfile variable.

The second command gets the SQL Server deployment object named SQL Deployment 01 from the SQL profile stored in \$SQLProfile and then stores the object in the \$SQLDeployment variable.

The last command removes the SQL Server deployment stored in \$SQLDeployment from the SQL Server profile stored in \$SQLProfile.

```
PS C:\> $SQLProfile = Get-SCSQLProfile -Name "SQLProfile01"
```

```
PS C:\> $SQLDeployment = Get-SCSQLDeployment -SQLProfile $SQLProfile -Name "SQL Deployment 01"
```

```
PS C:\> Remove-SCSQLDeployment -SQLDeployment $SQLDeployment
```

## Related topics

[Add-SCSQLDeployment](#)

[Get-SCSQLDeployment](#)

[Set-SCSQLDeployment](#)

---

# Remove-SCS SQLProfile

---

## Remove-SCS SQLProfile

Removes a SQL Server profile.

### Syntax

Parameter Set: Default

```
Remove-SCS SQLProfile [-SQLProfile] <SQLProfile> [-JobVariable <String> ] [-PROTipID <Guid> ]  
[-RunAsynchronously] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The Remove-SCS SQLProfile cmdlet removes a SQL Server profile from Virtual Machine Manager (VMM).

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-SQLProfile<SQLProfile>**

Specifies a SQL Server profile object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-Confirm**

Prompts you for confirmation before executing the command.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                             |       |
|-----------------------------|-------|
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## -WhatIf

Describes what would happen if you executed the command without actually executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Examples

### 1: Remove a SQL Server profile.

The first command gets the SQL Server profile object named SQLProfile01 and stores the object in the \$SQLProfile variable.

The second command removes the SQL Server profile object stored in \$SQLProfile from VMM.

```
PS C:\> $SQLProfile = Get-SCSQLProfile -Name "SQLProfile01"
```

```
PS C:\> Remove-SCSQLProfile -SQLProfile $SQLProfile
```

## Related topics

[Get-SCSQLProfile](#)

[New-SCSQLProfile](#)

[Set-SCSQLProfile](#)

---

# Remove-SCSQLScriptCommand

---

## Remove-SCSQLScriptCommand

Removes a SQL Server script from an application deployment.

### Syntax

Parameter Set: Default

```
Remove-SCSQLScriptCommand [-SQLScriptCommand] <SCSQLScriptCommand> [-JobVariable <String> ]  
[-PROTipID <Guid> ] [-RunAsynchronously] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The Remove-SCSQLScriptCommand cmdlet removes a SQL Server script from an application deployment.

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|



|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-SQLScriptCommand<SCSQLScriptCommand>**

Specifies a SQL Server script command object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-Confirm**

Prompts you for confirmation before executing the command.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                             |       |
|-----------------------------|-------|
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## -WhatIf

Describes what would happen if you executed the command without actually executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Examples

### 1: Remove a SQL Script script from an application deployment.

The first command gets the application profile object named SvcWebAppProfile01 and stores the object in the \$AppProfile variable.

The second command gets the application deployment object named SQLDataTierApp01 for the application profile stored in \$ApplicationProfile, and then stores the object in the \$AppDeployment variable.

The third command gets the first PreInstall SQL script (deployment order 1, SQL script type PreInstall) associated with the application deployment stored in \$AppDeployment.

The last command removes the SQL Server script stored in \$SQLScript.

```
PS C:\> $AppProfile = Get-SCApplicationProfile -Name "SvcWebAppProfile01"
```

```
PS C:\> $AppDeployment = Get-SCApplicationDeployment -ApplicationProfile $AppProfile -Name "SQLDataTierApp01"
```

```
PS C:\> $SQLScript = Get-SCSQLScriptCommand -ApplicationDeployment $AppDeployment | where {$_ .DeploymentOrder -eq "1" -and $_.SQLScriptType -eq "PreInstall"}
```

```
PS C:\> Remove-SCSQLScriptCommand -SQLScriptCommand $SQLScript
```

---

## Related topics

[Add-SCSQLScriptCommand](#)

[Get-SCSQLScriptCommand](#)

[Set-SCSQLScriptCommand](#)

---

# Remove-SCSSASConnection

---

## Remove-SCSSASConnection

Removes the SQL Server Analysis Services connection object from VMM.

### Syntax

Parameter Set: Default

```
Remove-SCSSASConnection [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Remove-SCSSASConnection cmdlet removes the SQL Server Analysis Services (SSAS) connection object from Virtual Machine Manager (VMM).

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

---

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **String**

## Examples

### **1: Remove the existing SQL Server Analysis Services connection.**

This command removes the SQL Server Analysis Services connection object from VMM.

```
PS C:\> Remove-SCSSASConnection
```

## Related topics

[Get-SCSSASConnection](#)

[New-SCSSASConnection](#)

---

# Remove-SCSSHKey

---

## Remove-SCSSHKey

Removes a Linux Administrator SSHkey object from VMM.

### Syntax

Parameter Set: Default

```
Remove-SCSSHKey [-SSHKey] <SSHKey> [-Force] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The Remove-SCSSHKey cmdlet removes a Linux Administrator SSHKey object from Virtual Machine Manager (VMM).

### Parameters

#### -Force

Forces the operation to complete.

For example:

- Remove-SCSCVMHost -Force

Forces the removal of a host object from the VMM database.

- Stop-SCVirtualMachine -Force

Stops a virtual machine.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |



---

## **-SSHKey<SSHKey>**

Specifies a Linux Administrator SSHKey object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-Confirm**

Prompts you for confirmation before executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Examples

### 1: Remove an SSHKey object.

The first command gets the SSHKey object named My.sshkey from the VMM library and stores the object in the \$SSHKey variable.

The second command removes the SSHKey object stored in \$SSHKey from VMM, prompting you for confirmation before performing the action.

```
PS C:\> $SSHKey = Get-SCSSHKey -Name "My.sshkey"
```

```
PS C:\> Remove-SCSSHKey -SSHKey $SSHKey -Confirm
```

## Related topics

[Get-SCSSHKey](#)

[Set-SCSSHKey](#)

---

# Remove-SCStaticIPAddressPool

---

## Remove-SCStaticIPAddressPool

Deletes a static IP address pool.

### Syntax

Parameter Set: Default

```
Remove-SCStaticIPAddressPool [-StaticIPAddressPool] <StaticIPAddressPool> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The Remove-SCStaticIPAddressPool cmdlet deletes a Virtual Machine Manager (VMM) static IP address pool.

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-StaticIPAddressPool<StaticIPAddressPool>**

Specifies an IP address pool from which you can assign static IP addresses.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-Confirm**

Prompts you for confirmation before executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- 
- **StaticIPAddressPool**

## Examples

### 1: Delete a static IP address pool for a specified subnet

The first command gets the host group with the path of All Hosts\HostGroup02\Production and stores it in the \$HostGroup variable.

The second command gets the static address pool named Production IP Address Pool for the host group stored in the \$HostGroup variable (including its parent host group if inheritance is enabled) using the IPv4 address for the specified subnet.

The last command deletes the static address pool stored in \$IPPool.

```
PS C:\> $HostGroup = Get-SCVMHostGroup | where { $_.Path -eq "All Hosts\HostGroup02\Production" }
```

```
PS C:\> $IPPool = Get-SCStaticIPAddressPool -IPv4 -Subnet "10.0.0.0/24" -VMHostGroup $Hostgroup -Name "Production IP Address Pool"
```

```
PS C:\> Remove-SCStaticIPAddressPool -StaticIPAddressPool $IPPool
```

## Related topics

[Get-SCStaticIPAddressPool](#)

[New-SCStaticIPAddressPool](#)

[Set-SCStaticIPAddressPool](#)

---

# Remove-SCStorageClassification

---

## Remove-SCStorageClassification

Deletes a storage classification object from the VMM database.

### Syntax

Parameter Set: Default

```
Remove-SCStorageClassification [-StorageClassification] <StorageClassification> [-  
JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

### Detailed Description

The Remove-SCStorageClassification cmdlet deletes a storage classification object from the Virtual Machine Manager (VMM) database. Deleting a storage classification removes the associations that storage classification had with any storage pools.

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-StorageClassification<StorageClassification>**

Specifies a storage classification object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-Confirm**

Prompts you for confirmation before executing the command.



|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Examples**

### **1: Remove a storage classification**

The first command gets the first item in the storage classification array and stores it in the \$Class variable.

The second command removes the storage classification stored in the \$Class variable.

```
PS C:\> $Class = @(Get-SCStorageClassification)[0]
```

```
PS C:\> Remove-SCStorageClassification -StorageClassification $Class
```

## **Related topics**

[Get-SCStorageClassification](#)

[New-SCStorageClassification](#)

[Set-SCStorageClassification](#)

---

# Remove-SCStorageFileShare

---

## Remove-SCStorageFileShare

Removes a storage file share from VMM.

### Syntax

Parameter Set: Default

```
Remove-SCStorageFileShare [-StorageFileShare] <StorageFileShare> [-ForceRemoveFileSystem] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

### Detailed Description

The Remove-SCStorageFileShare cmdlet removes a storage file share from Virtual Machine Manager (VMM).

### Parameters

#### -ForceRemoveFileSystem

Indicates that the storage file share is removed from the file system.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### -JobVariable<String>

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-StorageFileShare<StorageFileShare>**

Specifies a storage file share.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-Confirm**

Prompts you for confirmation before executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

---

## Examples

### 1: Remove a storage file share.

The first command gets the storage file share object named FileShare01 and stores the object in the \$FileShare variable.

The second command removes FileShare01 from VMM.

```
PS C:\> $FileShare = Get-SCStorageFileShare -Name "FileShare01"
```

```
PS C:\> Remove-SCStorageFileShare -StorageFileShare $FileShare
```

## Related topics

[Get-SCStorageFileShare](#)

[New-SCStorageFileShare](#)

[Register-SCStorageFileShare](#)

[Repair-SCStorageFileShare](#)

[Set-SCStorageFileShare](#)

[Unregister-SCStorageFileShare](#)

---

# Remove-SCStorageLogicalUnit

---

## Remove-SCStorageLogicalUnit

Deletes any associations a logical unit has to a host under VMM management.

### Syntax

Parameter Set: Default

```
Remove-SCStorageLogicalUnit [-StorageLogicalUnit] <StorageLogicalUnit> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The Remove-SCStorageLogicalUnit cmdlet removes any associations a logical unit has to a host under Virtual Machine Manager (VMM) management. The logical unit information remains in the VMM database. By default, this operation is not destructive. Optionally, Remove-SCStorageLogicalUnit can delete the logical unit instance from the storage pool, deleting all data contained.

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-StorageLogicalUnit<StorageLogicalUnit>**

Specifies a storage logical unit object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-Confirm**

Prompts you for confirmation before executing the command.

---

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## -WhatIf

Describes what would happen if you executed the command without actually executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Examples

### 1: Remove a storage logical unit

The first command gets the storage logical unit object named LUN01 and stores the object in the \$LU variable.

The second command deletes LUN01.

```
PS C:\> $LU = Get-SCStorageLogicalUnit -Name "LUN01"
```

```
PS C:\> Remove-SCStorageLogicalUnit -StorageLogicalUnit $LU
```

## Related topics

[Get-SCStorageLogicalUnit](#)

[New-SCStorageLogicalUnit](#)



---

[Register-SCStorageLogicalUnit](#)

[Set-SCStorageLogicalUnit](#)

[Unregister-SCStorageLogicalUnit](#)

---

# Remove-SCStorageProvider

---

## Remove-SCStorageProvider

Removes a storage provider from VMM.

### Syntax

Parameter Set: Default

```
Remove-SCStorageProvider [-StorageProvider] <StorageProvider> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The Remove-SCStorageProvider cmdlet removes a storage provider from Virtual Machine Manager (VMM). Removing a storage provider also removes any associated array, pool, and logical unit information exposed by the provider. However, this operation will not change or delete any data on the logical units.

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-StorageProvider<StorageProvider>**

Specifies a storage provider object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-Confirm**

Prompts you for confirmation before executing the command.

---

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Examples**

### **1: Remove a storage provider.**

The first command gets the storage provider named StorProv01 and stores it in the \$Provider variable.

The second command deletes the storage provider stored in the \$Provider variable.

```
PS C:\> $Provider = Get-SCStorageProvider -Name "StorProv01.Contoso.com"
```

```
PS C:\> Remove-SCStorageProvider -StorageProvider $Provider
```

## **Related topics**

[Add-SCStorageProvider](#)

[Get-SCStorageProvider](#)

[Read-SCStorageProvider](#)

[Set-SCStorageProvider](#)

---

# Remove-SCUpdateServer

---

## Remove-SCUpdateServer

Removes a Windows Server Update Services computer from VMM.

### Syntax

Parameter Set: Default

```
Remove-SCUpdateServer [-UpdateServer] <UpdateServer> -Credential <VMMCredential> [-  
JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer  
<ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Remove-SCUpdateServer cmdlet removes a Microsoft Windows Server Update Services (WSUS) computer from Virtual Machine Manager (VMM). Removing WSUS integration disables the update management feature in VMM.

### Parameters

#### **-Credential<VMMCredential>**

Specifies a credential object or, for some cmdlets, a Run As account object that contains the user name and password of an account that has permission to perform this action. Or, in the case of Restart-SCJob, has permission to complete a restarted task.

For more information about the PSCredential object, type: "Get-Help Get-Credential".

For more information about Run As accounts, type: "Get-Help New-SCRunAsAccount".

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-UpdateServer<UpdateServer>**

Specifies a VMM update server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **UpdateServer**

---

## Examples

### 1: Remove an update server.

The first command uses the Get-Credential cmdlet to prompt you to supply a user name and password and stores the credentials in the \$Credential variable.

The second command gets the update server object named WSUSComputer01 and stores the object in the \$UpdateServer variable.

The last command removes the update server stored in \$UpdateServer, in this case, WSUSComputer01.

```
PS C:\> $Credential = Get-SCRunAsAccount -Name "RunAsAccount01"
```

```
PS C:\> $UpdateServer = Get-SCUpdateServer -ComputerName "WSUSComputer01"
```

```
PS C:\> Remove-SCUpdateServer -UpdateServer $UpdateServer -Credential $Credential
```

## Related topics

[Add-SCUpdateServer](#)

[Get-SCUpdateServer](#)

[Set-SCUpdateServer](#)



---

# Remove-SCUplinkPortProfileSet

---

## Remove-SCUplinkPortProfileSet

Removes an uplink port profile set.

### Syntax

Parameter Set: Default

```
Remove-SCUplinkPortProfileSet [-UplinkPortProfileSet] <UplinkPortProfileSet> [-JobGroup <Guid> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The Remove-SCUplinkPortProfileSet cmdlet deletes an uplink port profile set.

### Parameters

#### **-JobGroup<Guid>**

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | false |

|                             |       |
|-----------------------------|-------|
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-UplinkPortProfileSet<UplinkPortProfileSet>**

Specifies an uplink port profile set object.

To obtain an uplink port profile set object, use the Get-SCUplinkPortProfileSet cmdlet.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

|                             |                |
|-----------------------------|----------------|
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-Confirm**

Prompts you for confirmation before executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

|           |       |
|-----------|-------|
| Required? | false |
| Position? | named |

---

|                             |       |
|-----------------------------|-------|
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Examples

### 1: Remove an uplink port profile set.

This command gets the uplink port profile set object named UplinkPortProfileSet01 and uses the pipeline operator to send the object to Remove-SCUplinkPortProfileSet which deletes the uplink port profile set object.

```
PS C:\> Get-SCUplinkPortProfileSet -Name "UplinkPortProfileSet01" | Remove-SCUplinkPortProfileSet
```

## Related topics

[Get-SCUplinkPortProfileSet](#)

[New-SCUplinkPortProfileSet](#)

[Set-SCUplinkPortProfileSet](#)

---

# Remove-SCUserRole

---

## Remove-SCUserRole

Removes an existing user role.

### Syntax

Parameter Set: Default

```
Remove-SCUserRole [-UserRole] <UserRole> [-JobGroup <Guid> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

### Detailed Description

The Remove-SCUserRole cmdlet removes an existing user role from Virtual Machine Manager (VMM). You can remove the following roles: Delegated Administrator, Read-Only Administrator, or Self-Service User. However, you cannot remove the Administrator role.

### Parameters

#### **-JobGroup<Guid>**

Specifies an identifier for a series of commands that will run as a set just before the final command that includes the same job group identifier runs.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

---

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-UserRole<UserRole>**

Specifies a user role object.

---

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Examples

### 1: Remove the specified user role.

The first command gets the user role object named ContosoDelegatedAdmin and stores the object in the \$UserRole variable.

The second command removes the "Contoso Admin" user role from the VMM database.

```
PS C:\> $UserRole = Get-SCUserRole -Name "ContosoDelegatedAdmin"
```

```
PS C:\> Remove-SCUserRole -UserRole $UserRole
```

## Related topics

[Get-SCUserRole](#)

[New-SCUserRole](#)

[Set-SCUserRole](#)

---

# Remove-SCVirtualDiskDrive

---

## Remove-SCVirtualDiskDrive

Removes a virtual disk drive object from a virtual machine or from a virtual machine template.

### Syntax

Parameter Set: Default

```
Remove-SCVirtualDiskDrive [-VirtualDiskDrive] <VirtualDiskDrive> [-Force] [-JobGroup <Guid> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-SkipDeleteVHD] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The Remove-SCVirtualDiskDrive cmdlet removes one or more virtual disk drive objects from a virtual machine or from a virtual machine template in a Virtual Machine Manager (VMM) environment.

### Parameters

#### -Force

Forces the operation to complete.

For example:

- Remove-SCSCVMHost -Force

Forces the removal of a host object from the VMM database.

- Stop-SCVirtualMachine -Force

Stops a virtual machine.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |



---

## **-JobGroup<Guid>**

Specifies an identifier for a series of commands that will run as a set just before the final command that includes the same job group identifier runs.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-SkipDeleteVHD**

Indicates that the VHD file will not be deleted when the virtual disk drive is removed.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VirtualDiskDrive<VirtualDiskDrive>**

Specifies a virtual disk drive object. You can attach either a virtual hard disk (for a virtual machine on any host) or a pass-through disk (for a virtual machine on a Hyper-V host or an ESX host) to a virtual disk drive object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

---

## **-Confirm**

Prompts you for confirmation before executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Notes**

- Requires a VMM virtual disk drive object, which can be retrieved by using the Get-SCVirtualDiskDrive cmdlet.

---

## Examples

### 1: Remove the second virtual disk drive object from the specified virtual machine.

The first command gets the virtual machine object named VM01 deployed on VMHost01 and stores the object in the \$VM variable.

The second command gets all virtual disk drive objects on VM01 and stores the retrieved objects in \$VirtDiskDrive. Using the '@' symbol and parentheses ensures that the command stores the results in an array in case the command returns a single object or a null value.

The last command returns the number of virtual disk drives associated with the virtual machine and then, if more than one exists, the command removes the second virtual disk drive (designated by the [1]) from the virtual machine.

```
PS C:\> $VM = Get-SCVirtualMachine | where { $_.VMHost.Name -eq "VMHost01.Contoso.com" -and $_.Name -eq "VM01" }
```

```
PS C:\> $VirtDiskDrive = @(Get-SCVirtualDiskDrive -VM $VM)
```

```
PS C:\> if($VirtDiskDrive.Count -gt 1){Remove-SCVirtualDiskDrive -VirtualDiskDrive $VirtDiskDrive[1]}
```

### 2: Remove all pass-through disks attached to a virtual machine.

The first command gets the virtual machine object named VM02 and stores the object in the \$VM variable.

The second command gets all virtual disk drive objects attached to VM02 that are not virtual hard disks (that is, only objects that represent pass-through disks are retrieved) and stores the pass-through disk objects in the \$VDDs object array.

The last command uses an if statement to determine whether at least one pass-through virtual disk drive exists. If the result is one or more, the command then uses the foreach statement to remove each virtual disk drive from the object array. Using the Force parameter ensures the removal of each virtual disk drive from its virtual machine even if other VMM objects depend on that virtual disk drive.

NOTE: For more information about the standard Windows PowerShell foreach loop statement, type: "Get-Help about\_ForEach".

```
PS C:\> $VM = Get-SCVirtualMachine | where { $_.Name -eq "VM02" }
```

```
PS C:\> $VirtDiskDrives = @(Get-SCVirtualDiskDrive -VM $VM | where { $_.IsVHD -eq $False })
```

```
PS C:\> if($VirtDiskDrives.Count -gt 0){foreach($VirtDiskDrive in $VirtDiskDrives){Remove-SCVirtualDiskDrive -Force -VirtualDiskDrive $VirtDiskDrive}}
```

### 3: Remove virtual disk drives attached to a virtual machine by name.

The first command gets all virtual machine objects whose name matches the string "WebSrvLOB" and stores the objects in the \$VM object array.

---

The second command uses the `ForEach` cmdlet to iterate through the virtual machines stored in `$VM` to get all virtual disk drive objects from each virtual machine. The command stores the virtual disk drive objects in the `$VirtDiskDrives` object array. Then, the command uses a second `ForEach` loop to select all virtual disk drive objects whose name contains the string `LOBData` from the `$VirtDiskDrives` array and passes these objects to the `Remove-SCVirtualDiskDrive` cmdlet which removes the objects from VMM.

```
PS C:\> $VMs = @(Get-SCVirtualMachine | where {$_.Name -match "WebSrvLOB"})
PS C:\> foreach ($VM in $VMs){$VirtDiskDrives = Get-SCVirtualDiskDrive -VM $VM; foreach
($VirtDiskDrive in $VirtDiskDrives){if($VirtDiskDrive.Name -match "LOBData"){Remove-
SCVirtualDiskDrive -VirtualDiskDrive $VirtDiskDrive}}}
```

## Related topics

[Convert-SCVirtualDiskDrive](#)

[Get-SCVirtualDiskDrive](#)

[New-SCVirtualDiskDrive](#)

[Set-SCVirtualDiskDrive](#)

---

# Remove-SCVirtualDVDDrive

---

## Remove-SCVirtualDVDDrive

Removes a virtual DVD drive object from VMM.

### Syntax

Parameter Set: SourceBusAndLunSpecified

```
Remove-SCVirtualDVDDrive -JobGroup <Guid> -SourceBus <Byte> -SourceLUN <Byte> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-Confirm] [-WhatIf] [<CommonParameters>]
```

Parameter Set: VirtualDVDDriveSpecified

```
Remove-SCVirtualDVDDrive [-VirtualDVDDrive] <VirtualDVDDrive> [-JobGroup <Guid> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-Confirm] [-WhatIf] [<CommonParameters>]
```

### Detailed Description

Removes one or more virtual DVD drive objects from a hardware profile, a virtual machine, or a virtual machine template used in a Virtual Machine Manager (VMM) environment. The cmdlet also deletes any .iso file that the virtual DVD drive uses from the file system on the library server.

This cmdlet returns the object upon success (with the property MarkedForDeletion set to TRUE) or returns an error message upon failure.

### Parameters

#### **-JobGroup<Guid>**

Specifies an identifier for a series of commands that will run as a set just before the final command that includes the same job group identifier runs.

|                        |       |
|------------------------|-------|
| Aliases                | none  |
| Required?              | false |
| Position?              | named |
| Default Value          | none  |
| Accept Pipeline Input? | false |

---

|                             |       |
|-----------------------------|-------|
| Accept Wildcard Characters? | false |
|-----------------------------|-------|

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                        |       |
|------------------------|-------|
| Aliases                | none  |
| Required?              | false |
| Position?              | named |
| Default Value          | none  |
| Accept Pipeline Input? | false |

---

|                             |       |
|-----------------------------|-------|
| Accept Wildcard Characters? | false |
|-----------------------------|-------|

### **-SourceBus<Byte>**

Specifies the source IDE bus for the drive.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-SourceLUN<Byte>**

Specifies the source logical unit number (LUN) for a virtual DVD drive object on an IDE bus.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VirtualDVDDrive<VirtualDVDDrive>**

Specifies a virtual DVD drive object.

|                        |                |
|------------------------|----------------|
| Aliases                | none           |
| Required?              | true           |
| Position?              | 1              |
| Default Value          | none           |
| Accept Pipeline Input? | True (ByValue) |



---

|                             |       |
|-----------------------------|-------|
| Accept Wildcard Characters? | false |
|-----------------------------|-------|

## **-Confirm**

Prompts you for confirmation before executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## **Notes**

- Requires a VMM virtual DVD drive object, which can be retrieved by using the Get-SCVirtualDVDDrive cmdlet.

---

## Examples

### 1: Remove a specific virtual DVD drive from a virtual machine.

The first command gets the virtual machine object named VM01 and stores the object in the \$VM variable.

The second command gets the virtual DVD drive object located on the first slot of the Secondary Channel (specified by -Bus 1 and -LUN 0) on the IDE bus on VM01 and then stores the virtual DVD drive object in the \$DVDDrive variable.

The last command removes the virtual DVD drive object stored in \$DVDDrive from VM01 and deletes any .iso file that this virtual DVD drive uses from the file system on the library server.

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM01"
PS C:\> $DVDDrive = Get-SCVirtualDVDDrive -VM $VM | where { $_.Bus -eq 1 -and $_.LUN -eq 0 }
PS C:\> Remove-SCVirtualDVDDrive -VirtualDVDDrive $DVDDrive
```

### 2: Remove the third virtual DVD drive from a virtual machine.

The first command gets the virtual machine object named VM02 and stores the object in the \$VM variable.

The second command gets all virtual DVD drive objects connected to VM02 and stores each virtual DVD drive object in the \$DVDDrive object array. This example assumes that VM02 has three virtual DVD drives and therefore the array contains three elements (counting 0 to 2).

The last command passes the third virtual DVD drive (object [2]) stored in \$DVDDrive to the Remove-SCVirtualDVDDrive cmdlet, which removes this virtual DVD drive object from VM02 and deletes any .iso file used by this virtual DVD drive from the file system on the library server.

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM02"
PS C:\> $DVDDrive = Get-SCVirtualDVDDrive -VM $VM
PS C:\> $DVDDrive[2] | Remove-SCVirtualDVDDrive
```

## Related topics

[Get-SCVirtualDVDDrive](#)

[New-SCVirtualDVDDrive](#)

[Set-SCVirtualDVDDrive](#)

---

# Remove-SCVirtualFloppyDisk

---

## Remove-SCVirtualFloppyDisk

Removes a virtual floppy disk object from VMM.

### Syntax

Parameter Set: Default

```
Remove-SCVirtualFloppyDisk [-VirtualFloppyDisk] <VirtualFloppyDisk> [-Force] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

### Detailed Description

The Remove-SCVirtualFloppyDisk cmdlet removes a virtual floppy disk object from the VMM library and deletes the corresponding virtual floppy disk file (a Windows-based .vfd file or a VMware-based .flp file) from the library server.

If the virtual floppy disk is attached to a virtual machine, template, or hardware profile (and if you do not use the Force parameter), Virtual Machine Manager (VMM) lists the container that contains the virtual floppy disk and prompts you to confirm that you want to remove the virtual floppy disk:

- If you reply Yes, VMM removes the association between the virtual floppy disk and the container to which it is attached, and then deletes the virtual floppy disk object from VMM.

- If you reply No, the operation is cancelled.

This cmdlet returns the object upon success (with the property MarkedForDeletion set to TRUE) or returns an error message upon failure.

### Parameters

#### -Force

Forces the operation to complete.

For example:

```
- Remove-SCSCVMHost -Force
```

Forces the removal of a host object from the VMM database.

```
- Stop-SCVirtualMachine -Force
```

Stops a virtual machine.

---

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VirtualFloppyDisk<VirtualFloppyDisk>**

Specifies a virtual floppy disk object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Notes**

- Requires a VMM virtual floppy disk object, which can be retrieved by using the Get-SCVirtualFloppyDisk cmdlet.

## **Examples**

### **1: Remove a virtual floppy disk object from the library and delete the corresponding file.**

The first command gets the virtual floppy disk file object named BootFloppy.vfd stored on LibraryServer01, and then stores the virtual floppy disk object in the \$VFD variable.

The second command removes the floppy disk object stored in \$VFD from the library and deletes the corresponding virtual floppy disk file from the library server.

---

```
PS C:\> $VFD = Get-SCVirtualFloppyDisk -VMMServer "VMMServer01.Contoso.com" | where {
$_Name -eq "BootFloppy.vfd" -and $_.LibraryServer.Name -eq "LibraryServer01.Contoso.com" }
PS C:\> Remove-SCVirtualFloppyDisk -VirtualFloppyDisk $VFD
```

## 2: Remove multiple virtual floppy disks and their files.

The first command gets all virtual floppy disk objects whose names include the string "Boot" and stores these objects in the array named \$VFDs.

The second command passes each virtual floppy disk object in \$VFDs to the Remove-VirtualFloppyDisk cmdlet, which removes each virtual floppy disk object from the library. The command also deletes each corresponding file from the library server on which that virtual floppy disk is stored.

```
PS C:\> $VFDs = Get-SCVirtualFloppyDisk -VMMServer "VMMServer01.Contoso.com" | where {
$_Name -match "Boot" }
PS C:\> $VFDs | Remove-SCVirtualFloppyDisk
```

## Related topics

[Get-SCVirtualFloppyDisk](#)

[Set-SCVirtualFloppyDisk](#)

---

# Remove-SCVirtualHardDisk

---

## Remove-SCVirtualHardDisk

Removes a virtual hard disk object from a virtual machine or template, or from the VMM library.

### Syntax

Parameter Set: Default

```
Remove-SCVirtualHardDisk [-VirtualHardDisk] <VirtualHardDisk> [-Force] [-JobGroup <Guid> ]  
[-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

### Detailed Description

The Remove-SCVirtualHardDisk cmdlet removes a virtual hard disk object from a virtual machine or template, or from the VMM library. Remove-VirtualHardDisk also deletes the corresponding virtual hard disk file (a Windows-based .vhd file, a Citrix XenServer-based .vhd file, or a VMware-based .vmdk file) from the library server.

If the virtual hard disk is attached to a virtual disk drive on a virtual machine or template (and if you do not use the Force parameter), Virtual Machine Manager (VMM) lists the container that contains the virtual hard disk and prompts you to confirm that you want to remove the virtual hard disk:

- If you reply Yes, VMM removes the association between the virtual hard disk and the container to which it is attached, and then deletes the virtual hard disk object from VMM.

- If you reply No, the operation is cancelled.

This cmdlet returns the object upon success (with the property MarkedForDeletion set to TRUE) or returns an error message upon failure.

### Parameters

#### -Force

Forces the operation to complete.

For example:

- Remove-SCSCVMHost -Force

Forces the removal of a host object from the VMM database.

- Stop-SCVirtualMachine -Force

---

Stops a virtual machine.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobGroup<Guid>**

Specifies an identifier for a series of commands that will run as a set just before the final command that includes the same job group identifier runs.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |



---

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VirtualHardDisk<VirtualHardDisk>**

Specifies a virtual hard disk object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

---

## -Confirm

Prompts you for confirmation before executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## -WhatIf

Describes what would happen if you executed the command without actually executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## Notes

- Requires a VMM virtual hard disk object, which can be retrieved by using the Get-SCVirtualHardDisk cmdlet.

## Examples

### 1: Remove a virtual hard disk object from the library.

The first command gets the virtual hard disk object named VHD01.vhd stored on LibraryServer01 and stores the returned object in the \$VHD variable.

The second command removes the virtual hard disk object stored in \$VHD from the library and deletes the corresponding file from the file system on the library server.

---

```
PS C:\> $VHD = Get-SCVirtualHardDisk -VMMServer "VMMServer01.Contoso.com" | where { $_.Name -eq "VHD01.vhd" -and $_.LibraryServer.Name -eq "LibraryServer01.Contoso.com" }
PS C:\> Remove-SCVirtualHardDisk -VirtualHardDisk $VHD
```

## 2: Remove a virtual hard disk from a virtual machine.

The first command connects to VMMServer01.

The second command gets the virtual machine object named VM01, gets all virtual hard disks on VM01 whose name includes the string "DataDisk", and then stores these virtual hard disk objects in an array named \$VHD.

The third command removes each virtual hard disk object stored in the \$VHD array from the virtual machine and deletes each corresponding file from the file system on the library server.

```
PS C:\> Get-SCVMMServer -ComputerName "VMMServer01.Contoso.com"
PS C:\> $VHD = Get-SCVirtualMachine -Name "VM01" | Get-SCVirtualHardDisk | where { $_.Name -match "DataDisk" }
PS C:\> $VHD | Remove-SCVirtualHardDisk
```

## Related topics

[Get-SCVirtualHardDisk](#)

[Move-SCVirtualHardDisk](#)

[Set-SCVirtualHardDisk](#)

---

# Remove-SCVirtualizationManager

---

## Remove-SCVirtualizationManager

Removes a VMware vCenter Server from VMM.

### Syntax

Parameter Set: Default

```
Remove-SCVirtualizationManager [-VirtualizationManager] <VirtualizationManager> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

### Detailed Description

The Remove-SCVirtualizationManager cmdlet removes one or more VMware vCenter Server objects from Virtual Machine Manager (VMM). This cmdlet deletes the vCenter Server object from the VMM database and also removes all imported ESX host objects and virtual machine objects associated with the vCenter Server.

When you remove a VirtualCenter Server, the cmdlet does not make any changes within the vCenter Server and does not remove any hosts or virtual machines from the vCenter Server.

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VirtualizationManager<VirtualizationManager>**

Specifies a virtualization manager object managed by VMM.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Notes

- Requires a VMM virtualization manager object, which can be retrieved by using the Get-SCVirtualizationManager cmdlet.

## Examples

### 1: Remove a VMware vCenter Server from VMM.

The first command gets the virtualization manager object named VirtMgrServer01 from VMMServer01 and stores the object in the \$VirtMgrServer variable.

The second command removes the vCenter Server object, as well as all associated host and virtual machine objects, from VMM.

```
PS C:\> $VirtMgrServer = Get-SCVirtualizationManager -VMMServer "VMMServer01.Contoso.com" -
ComputerName "VirtMgrServer01.Contoso.com"
```

```
PS C:\> Remove-SCVirtualizationManager -VirtualizationManager $VirtMgrServer
```

### 2: Remove a set of VMware vCenter Servers from VMM.

The first command gets all virtualization manager objects whose name includes the string "Server" and stores the objects in \$VirtManagers.

The second command removes each object in \$VirtManagers from VMM, as well as all associated host and virtual machine objects.

For more information about the standard Windows PowerShell foreach loop statement, type: Get-Help about\_ForEach.

```
PS C:\> $VirtManagers = Get-SCVirtualizationManager -VMMServer "VMMServer01.Contoso.com" |
where { $_.Name -match "Server" }
```

```
PS C:\> ForEach ($VirtManager in $VirtManagers) {Remove-SCVirtualizationManager -
VirtualizationManager $VirtManager}
```

### 3: Remove all VMware vCenter Servers from VMM.

This command removes all virtualization manager objects from VMM.

```
PS C:\> Get-SCVirtualizationManager | Remove-SCVirtualizationManager -RunAsynchronously
```

---

## Related topics

[Get-SCVirtualizationManager](#)

[Add-SCVirtualizationManager](#)

[Read-SCVirtualizationManager](#)

[Set-SCVirtualizationManager](#)

---

# Remove-SCVirtualMachine

---

## Remove-SCVirtualMachine

Removes a virtual machine object from VMM.

### Syntax

Parameter Set: Default

```
Remove-SCVirtualMachine [-VM] <VM> [-Force] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

### Detailed Description

The Remove-SCVirtualMachine cmdlet removes a virtual machine object deployed on a host or stored on a VMM library server.

Remove-SCVirtualMachine deletes the virtual machine record from the Virtual Machine Manager (VMM) database, deletes all files associated with the virtual machine, and removes the virtual machine from the host on which it is deployed or from the library server on which it is stored.

If a folder on a host was created for this virtual machine by VMM (rather than by Hyper-V or VMware) and if that folder contains no other virtual machines or other data, you can use the file system to delete the folder after you have removed the virtual machine.

NOTE: When used with the Force parameter, Remove-SCVirtualMachine only deletes the virtual machine from the VMM database. It does not delete the virtual machine itself.

This cmdlet returns the object upon success (with the property MarkedForDeletion set to TRUE) or returns an error message upon failure.

### Parameters

#### -Force

Forces the operation to complete.

For example:

```
- Remove-SCSCVMHost -Force
```

Forces the removal of a host object from the VMM database.

```
- Stop-SCVirtualMachine -Force
```

Stops a virtual machine.



---

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VM<VM>**

Specifies a virtual machine object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Notes**

- Requires a VMM virtual machine object, which can be retrieved by using the Get-SCVirtualMachine cmdlet.

## **Examples**

### **1: Remove a specific virtual machine deployed on a host.**

The first command gets the virtual machine object named VM01 deployed on VMHost01 and then stores the virtual machine object in the \$VM variable.

The second command removes the object stored in \$VM and deletes the corresponding virtual machine files from the file system on its host.

---

```
PS C:\> $VM = Get-SCVirtualMachine -VMMServer "VMMServer01.Contoso.com" | where {
$_ .VMHost.Name -eq "VMHost01.Contoso.com" -and $_.Name -eq "VM01" }
PS C:\> Remove-SCVirtualMachine -VM $VM
```

## 2: Remove all virtual machines with names that include a specific string.

The first command gets all virtual machine objects deployed on any host whose name includes the string "VM0" and then stores these virtual machine objects in the array named \$VMs.

The second command removes each virtual machine object in the \$VMs array and deletes the corresponding virtual machine files from the file system on each host.

```
PS C:\> $VMs = @(Get-SCVirtualMachine -VMMServer "VMMServer01.Contoso.com" | where { $_.Name
-Match "VM0" } )
PS C:\> $VMs | Remove-SCVirtualMachine
```

## 3: Remove a specific virtual machine stored on a VMM library server.

The first command gets the object that represents the virtual machine named VM03 (which is stored on the library server named FileServer01) and stores the virtual machine object in \$VM. This example assumes that only one virtual machine named VM03 exists.

The second command removes the object that represents VM03 from the library and deletes the corresponding virtual machine files from the file system on the library server.

```
PS C:\> $VM = Get-SCVirtualMachine -VMMServer "VMMServer1.Contoso.com" | where {
$_ .LibraryServer.Name -eq "LibraryServer01.Contoso.com" -and $_.Name -eq "VM02" }
PS C:\> Remove-SCVirtualMachine -VM $VM
```

## 4: Remove multiple stored virtual machines from the VMM library.

The first command gets all virtual machine objects whose names include the string "VM0" and that are stored on LibraryServer01. The command then stores the virtual machine objects in the array named \$VMs.

The second command passes each virtual machine object stored in \$VMs to the Remove-SCVirtualMachine cmdlet, which removes each object from the library and deletes the corresponding virtual machine files from the file system on the library server. The Confirm parameter prompts you to confirm whether you want to delete each of these virtual machines.

```
PS C:\> $VMs = Get-SCVirtualMachine -VMMServer "VMMServer01.Contoso.com" | where {
$_ .LibraryServer.Name -eq "LibraryServer01.Contoso.com" -and $_.Name -match "VM0" }
PS C:\> $VMs | Remove-SCVirtualMachine -Confirm
```

---

## Related topics

[Get-SCVirtualMachine](#)

[Move-SCVirtualMachine](#)

[New-SCVirtualMachine](#)

[Read-SCVirtualMachine](#)

[Repair-SCVirtualMachine](#)

[Resume-SCVirtualMachine](#)

[Save-SCVirtualMachine](#)

[Set-SCVirtualMachine](#)

[Start-SCVirtualMachine](#)

[Stop-SCVirtualMachine](#)

[Suspend-SCVirtualMachine](#)

---

# Remove-SCVirtualNetwork

---

## Remove-SCVirtualNetwork

Removes a virtual network from a host managed by VMM.

### Syntax

Parameter Set: Host

```
Remove-SCVirtualNetwork [-VirtualNetwork] <VirtualNetwork> [-JobGroup <Guid> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: Cluster

```
Remove-SCVirtualNetwork [-ClusterVirtualNetwork] <ClusterVirtualNetwork> [-JobGroup <Guid> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

### Detailed Description

The Remove-SCVirtualNetwork cmdlet removes one or more virtual network objects from Virtual Machine Manager (VMM).

This cmdlet returns the object upon success (with the property MarkedForDeletion set to True) or returns an error message upon failure.

### Parameters

#### **-ClusterVirtualNetwork<ClusterVirtualNetwork>**

Specifies a cluster virtual network object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

---

### **-JobGroup<Guid>**

Specifies an identifier for a series of commands that will run as a set just before the final command that includes the same job group identifier runs.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                        |       |
|------------------------|-------|
| Aliases                | none  |
| Required?              | false |
| Position?              | named |
| Default Value          | none  |
| Accept Pipeline Input? | false |

---

|                             |       |
|-----------------------------|-------|
| Accept Wildcard Characters? | false |
|-----------------------------|-------|

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VirtualNetwork<VirtualNetwork>**

Specifies a virtual network object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-Confirm**

Prompts you for confirmation before executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## -WhatIf

Describes what would happen if you executed the command without actually executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Examples

### 1: Remove a specific virtual network from a host.

The first command gets the host object named VMHost01 and stores the object in the \$VMHost variable.

The second command gets the virtual network object named InternalVNet01 configured on VMHost01 and stores the object in the \$Network variable.

The last command removes InternalVNet01 from VMHost01, prompting you for confirmation before continuing the action.

```
PS C:\> $VMHost = Get-SCVMHost -ComputerName "VMHost01.Contoso.com"
```

```
PS C:\> $Network = Get-SCVirtualNetwork -VMHost $VMHost -Name "InternalVNet01"
```

```
PS C:\> Remove-SCVirtualNetwork -VirtualNetwork $Network -Confirm
```

## Related topics

[Get-SCVirtualNetwork](#)

[New-SCVirtualNetwork](#)

[Set-SCVirtualNetwork](#)



---

# Remove-SCVirtualNetworkAdapter

---

## Remove-SCVirtualNetworkAdapter

Removes a virtual network adapter object from VMM.

### Syntax

Parameter Set: SlotIdSpecified

```
Remove-SCVirtualNetworkAdapter -JobGroup <Guid> -SlotID <Int32> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: VirtualNicSpecified

```
Remove-SCVirtualNetworkAdapter [-VirtualNetworkAdapter] <VirtualNetworkAdapter> [-JobGroup <Guid> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The Remove-SCVirtualNetworkAdapter cmdlet removes one or more virtual network adapter objects from a virtual machine, virtual machine template, or hardware profile used in a Virtual Machine Manager (VMM) environment.

This cmdlet returns the object upon success (with the property MarkedForDeletion set to TRUE) or returns an error message upon failure.

### Parameters

#### **-JobGroup<Guid>**

Specifies an identifier for a series of commands that will run as a set just before the final command that includes the same job group identifier runs.

|                        |       |
|------------------------|-------|
| Aliases                | none  |
| Required?              | false |
| Position?              | named |
| Default Value          | none  |
| Accept Pipeline Input? | false |

---

|                             |       |
|-----------------------------|-------|
| Accept Wildcard Characters? | false |
|-----------------------------|-------|

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                        |       |
|------------------------|-------|
| Aliases                | none  |
| Required?              | false |
| Position?              | named |
| Default Value          | none  |
| Accept Pipeline Input? | false |

|                             |       |
|-----------------------------|-------|
| Accept Wildcard Characters? | false |
|-----------------------------|-------|

### **-SlotID<Int32>**

Specifies a numerical ID used to identify a device.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VirtualNetworkAdapter<VirtualNetworkAdapter>**

Specifies a virtual network adapter object for a virtual machine.

TYPE OF HOST      NUMBER OF VIRTUAL NETWORK ADAPTERS

-----      -----

Hyper-V            Up to 4 emulated adapters per virtual machine.

Up to 8 synthetic adapters per virtual machine.

(Exception: no driver available for an emulated network adapter on a Windows Server 2003 x64 guest.)

VMware ESX      Up to 4 emulated adapters per virtual machine.

Citrix XenServer Up to 7 emulated adapters per virtual machine.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

---

## -Confirm

Prompts you for confirmation before executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## -WhatIf

Describes what would happen if you executed the command without actually executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## Notes

- Requires a VMM virtual network adapter object, which can be retrieved by using the Get-SCVirtualNetworkAdapter cmdlet.

## Examples

### 1: Remove a virtual network adapter with the specified MAC address from a virtual machine.

The first command gets the virtual machine object named VM01 and stores the object in the \$VM variable.

---

The second command gets the virtual network adapter object on VM01 that has the specified MAC address and stores the object in the \$Adapter variable.

The last command removes the virtual network adapter stored in \$Adapter from VM01.

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM01"
```

```
PS C:\> $Adapter = Get-SCVirtualNetworkAdapter -VM $VM | where { $_.PhysicalAddress -eq "00:16:D3:CC:00:1B" }
```

```
PS C:\> Remove-SCVirtualNetworkAdapter -VirtualNetworkAdapter $Adapter
```

## **2: Remove a virtual network adapter connected to a specific virtual network from a virtual machine.**

The first command gets the virtual machine object named VM02 and stores the object in the \$VM variable.

The second command gets the virtual network adapter object on VM02 that is connected to the specified virtual network and stores the object in the \$Adapter variable.

The last command removes the virtual network adapter object stored in \$Adapter from VM02.

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM02"
```

```
PS C:\> $Adapter = Get-SCVirtualNetworkAdapter -VM $VM | where { $_.VirtualNetwork -eq "ExternalVirtualNetwork01" }
```

```
PS C:\> Remove-SCVirtualNetworkAdapter -VirtualNetworkAdapter $Adapter
```

## **3: Remove the only virtual network adapter from a virtual machine.**

The first command gets the virtual machine object named VM03 and stores the object in the \$VM variable.

The second command gets the virtual network adapter object on VM03 and stores the object in the \$Adapter variable. This example assumes that VM03 has only one virtual network adapter.

The last command removes the virtual network adapter object stored in \$Adapter from VM03.

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM03"
```

```
PS C:\> $Adapter = Get-SCVirtualNetworkAdapter -VM $VM
```

```
PS C:\> Remove-SCVirtualNetworkAdapter -VirtualNetworkAdapter $Adapter
```

## **4: Remove all virtual network adapters from a virtual machine.**

The first command gets the virtual machine object named VM04 and stores the object in the \$VM variable.

The second command gets all virtual network adapter objects on VM04 and stores the objects in the \$Adapters object array.

The last command passes each object stored in \$Adapters to Remove-SCVirtualNetworkAdapter, which removes each virtual network adapter object from VM04.

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM04"
```

---

```
PS C:\> $Adapters = Get-SCVirtualNetworkAdapter -VM $VM
PS C:\> $Adapters | Remove-SCVirtualNetworkAdapter
```

## 5: Remove the second virtual network adapter from a virtual machine that has three virtual network adapters.

The first command gets the virtual machine object named VM05 and stores the object in the \$VM variable.

The second command gets all virtual network adapter objects on VM05 and stores the objects in the \$Adapters object array. This example assumes that VM05 has three virtual network adapters.

The last command passes the second virtual network adapter object (\$Adapters [1]) to the Remove-SCVirtualNetworkAdapter cmdlet, which removes this virtual network adapter object from VM05.

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM05"
PS C:\> $Adapters = Get-SCVirtualNetworkAdapter -VM $VM
PS C:\> $Adapters[1] | Remove-SCVirtualNetworkAdapter
```

### Related topics

[Get-SCVirtualNetworkAdapter](#)

[New-SCVirtualNetworkAdapter](#)

[Set-SCVirtualNetworkAdapter](#)

---

# Remove-SCVirtualNetworkAdapterNativePortProfile

---

## Remove-SCVirtualNetworkAdapterNativePortProfile

Removes a virtual network adapter native port profile.

### Syntax

Parameter Set: Default

```
Remove-SCVirtualNetworkAdapterNativePortProfile [-VirtualNetworkAdapterNativePortProfile]
<VirtualNetworkAdapterNativePortProfile> [-JobVariable <String> ] [-PROTipID <Guid> ] [-
RunAsynchronously] [-VMMServer <ServerConnection> ] [-Confirm] [-WhatIf] [
<CommonParameters>]
```

### Detailed Description

The Remove-SCVirtualNetworkAdapterNativePortProfile cmdlet deletes a virtual network adapter native port profile.

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

-

## **VirtualNetworkAdapterNativePortProfile<VirtualNetworkAdapterNativePortProfile>**

Specifies a virtual network adapter native port profile object.

To obtain a virtual network adapter native port profile object, use the Get-SCVirtualNetworkAdapterNativePortProfile cmdlet.

|               |      |
|---------------|------|
| Aliases       | none |
| Required?     | true |
| Position?     | 1    |
| Default Value | none |



|                             |                |
|-----------------------------|----------------|
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-Confirm**

Prompts you for confirmation before executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## Examples

### 1: Remove a virtual network adapter native port profile.

This command gets the virtual network adapter native port profile object named VirtualNetworkAdapterNativePortProf01 and passes the object to Remove-SCVirtualNetworkAdapterNativePortProfile which deletes the profile object.

```
PS C:\> Get-SCVirtualNetworkAdapterNativePortProfile -Name  
"VirtualNetworkAdapterNativePortProf01" | Remove-SCVirtualNetworkAdapterNativePortProfile
```

## Related topics

[Get-SCVirtualNetworkAdapterNativePortProfile](#)

[New-SCVirtualNetworkAdapterNativePortProfile](#)

[Set-SCVirtualNetworkAdapterNativePortProfile](#)

---

# Remove-SCVirtualNetworkAdapterPortProfileSet

---

## Remove-SCVirtualNetworkAdapterPortProfileSet

Removes a virtual network adapter port profile set.

### Syntax

Parameter Set: Default

```
Remove-SCVirtualNetworkAdapterPortProfileSet [-VirtualNetworkAdapterPortProfileSet]
<VirtualNetworkAdapterPortProfileSet> [-JobGroup <Guid> ] [-JobVariable <String> ] [-
PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [-Confirm] [-WhatIf]
[ <CommonParameters>]
```

### Detailed Description

The Remove-SCVirtualNetworkAdapterPortProfileSet cmdlet deletes a virtual network adapter port profile set.

### Parameters

#### **-JobGroup<Guid>**

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

---

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

-

## **VirtualNetworkAdapterPortProfileSet<VirtualNetworkAdapterPortProfileSet>**

Specifies a virtual network adapter port profile set object.

To obtain a virtual network adapter port profile set object, use the Get-SCVirtualNetworkAdapterPortProfileSet cmdlet.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-Confirm**

Prompts you for confirmation before executing the command.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Position?              | named |
| Default Value          | none  |
| Accept Pipeline Input? | false |

---

|                             |       |
|-----------------------------|-------|
| Accept Wildcard Characters? | false |
|-----------------------------|-------|

## -WhatIf

Describes what would happen if you executed the command without actually executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Examples

### 1: Remove a virtual network adapter port profile set.

This command gets the virtual network adapter port profile set named VirtualNetworkAdapterPortProfSet01 and uses the pipeline operator to pass the object to Remove-SCVirtualNetworkAdapterPortProfileSet which deletes the virtual network adapter port profile set object.

```
PS C:\> Get-SCVirtualNetworkAdapterPortProfileSet -Name "VirtualNetworkAdapterPortProfSet01"
| Remove-SCVirtualNetworkAdapterPortProfileSet
```

## Related topics

[Get-SCVirtualNetworkAdapterPortProfileSet](#)

[New-SCVirtualNetworkAdapterPortProfileSet](#)

[Set-SCVirtualNetworkAdapterPortProfileSet](#)

---

# Remove-SCVirtualScsiAdapter

---

## Remove-SCVirtualScsiAdapter

Removes a virtual SCSI adapter object from VMM.

### Syntax

Parameter Set: Default

```
Remove-SCVirtualScsiAdapter [-VirtualScsiAdapter] <VirtualSCSIAdapter> [-JobGroup <Guid> ]  
[-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

### Detailed Description

The Remove-SCVirtualScsiAdapter cmdlet removes one or more virtual SCSI adapter objects from a virtual machine, virtual machine template, or hardware profile used in a Virtual Machine Manager (VMM) environment.

The Remove-SCVirtualSCSIAdapter cmdlet removes a virtual SCSI adapter successfully only if the adapter does not have any devices attached to it.

A virtual machine on a Citrix XenServer host always has one virtual SCSI adapter. You cannot remove this adapter.

This cmdlet returns the object upon success (with the property MarkedForDeletion set to TRUE) or returns an error message upon failure.

### Parameters

#### -JobGroup<Guid>

Specifies an identifier for a series of commands that will run as a set just before the final command that includes the same job group identifier runs.

|                        |       |
|------------------------|-------|
| Aliases                | none  |
| Required?              | false |
| Position?              | named |
| Default Value          | none  |
| Accept Pipeline Input? | false |

---

|                             |       |
|-----------------------------|-------|
| Accept Wildcard Characters? | false |
|-----------------------------|-------|

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                        |       |
|------------------------|-------|
| Aliases                | none  |
| Required?              | false |
| Position?              | named |
| Default Value          | none  |
| Accept Pipeline Input? | false |



|                             |       |
|-----------------------------|-------|
| Accept Wildcard Characters? | false |
|-----------------------------|-------|

## **-VirtualScsiAdapter<VirtualSCSIAdapter>**

Specifies a virtual SCSI adapter object for a virtual machine.

TYPE OF HOST    NUMBER OF VIRTUAL SCSI ADAPTERS

-----    -----

Hyper-V        Up to 4 synthetic virtual SCSI adapters per VM,  
and up to 64 devices per adapter

Supports a virtual disk drive size up to 2040 GB.

Does not support emulated virtual SCSI adapters.

VMware ESX    Up to 4 virtual SCSI adapters per VM,  
and up to 15 devices per adapter.

Supports a virtual disk drive size up to 2048 GB.

Citrix XenServer Always 1 virtual SCSI adapter per VM,  
and up to 8 devices per adapter.

Supports a virtual disk drive size up to 2048 GB.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-Confirm**

Prompts you for confirmation before executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## -WhatIf

Describes what would happen if you executed the command without actually executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Notes

- Requires a VMM virtual SCSI adapter object, which can be retrieved by using the Get-VirtualSCSIAdapter cmdlet.

## Examples

### 1: Remove the third virtual SCSI adapter from a virtual machine.

The first command gets the virtual machine object named VM01 and stores the object in the \$VM variable.

The second command gets all virtual SCSI adapter objects on VM01 and stores the objects in the \$Adapter object array. A virtual machine can have up to four virtual SCSI adapters attached. This example assumes that VM01 has at least three virtual SCSI adapters.

The last command passes the third virtual SCSI adapter (\$Adapter[2]) to Remove-SCVirtualScsiAdapter, which removes this virtual SCSI adapter from VM01.

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM01"
PS C:\> $Adapter = Get-SCVirtualSCSIAdapter -VM $VM
PS C:\> $Adapter[2] | Remove-SCVirtualScsiAdapter
```

## Related topics

[Get-SCVirtualScsiAdapter](#)

[New-SCVirtualScsiAdapter](#)

[Set-SCVirtualScsiAdapter](#)

---

# Remove-SCVirtualSwitchExtensionManager

---

## Remove-SCVirtualSwitchExtensionManager

Removes a virtual switch extension manager from VMM.

### Syntax

Parameter Set: Default

```
Remove-SCVirtualSwitchExtensionManager [-VirtualSwitchExtensionManager]  
<VirtualSwitchExtensionManager> [-JobVariable <String> ] [-PROTipID <Guid> ] [-  
RunAsynchronously] [-VMMServer <ServerConnection> ] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

### Detailed Description

The Remove-SCVirtualSwitchExtensionManager cmdlet deletes a virtual switch extension manager from Virtual Machine Manager (VMM).

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VirtualSwitchExtensionManager<VirtualSwitchExtensionManager>**

Specifies a virtual switch extension manager object.

To obtain a virtual switch extension manager object, use the Get-SCVirtualSwitchExtensionManager cmdlet.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

---

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-Confirm**

Prompts you for confirmation before executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Related topics

[Add-SCVirtualSwitchExtensionManager](#)

[Get-SCVirtualSwitchExtensionManager](#)

[Read-SCVirtualSwitchExtensionManager](#)

[Set-SCVirtualSwitchExtensionManager](#)

[Test-SCVirtualSwitchExtensionManager](#)

---

# Remove-SCVMCheckpoint

---

## Remove-SCVMCheckpoint

Removes a virtual machine checkpoint object from the VMM database.

### Syntax

Parameter Set: Default

```
Remove-SCVMCheckpoint -VMCheckpoint <VMCheckpoint> [-JobGroup <Guid> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

### Detailed Description

The Remove-SCVMCheckpoint cmdlet removes a virtual machine checkpoint object from the Virtual Machine Manager (VMM) database.

### Parameters

#### **-JobGroup<Guid>**

Specifies an identifier for a series of commands that will run as a set just before the final command that includes the same job group identifier runs.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

---

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMCheckpoint<VMCheckpoint>**

Specifies a VMM virtual machine checkpoint object.



|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-Confirm**

Prompts you for confirmation before executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

---

## Notes

- Requires a VMM checkpoint object, which can be retrieved by using the Get-SCVMCheckpoint cmdlet.

## Examples

### 1: Remove the most recent checkpoint for a virtual machine.

The first command gets all checkpoint objects for VM01 and stores these objects in the \$Checkpoints object array. This example assumes that VM01 has several checkpoints.

The second command removes the first checkpoint in the array (\$Checkpoints[0]). The Confirm parameter prompts you to confirm whether you want to remove the checkpoint.

```
PS C:\> $Checkpoint = Get-VMCheckpoint -VM "VM01" -MostRecent
PS C:\> Remove-SCVMCheckpoint -VMCheckpoint $Checkpoint -Confirm
```

### 2: Remove a specified checkpoint for a virtual machine.

The first command gets all checkpoint objects for VM01 and stores the objects in the \$Checkpoints object array. This example assumes that VM01 has at least two checkpoints.

The second command removes the first checkpoint stored in the \$Checkpoints array, which is the first checkpoint created for VM01. The command prompts you for confirmation before proceeding.

```
PS C:\> $Checkpoints = Get-VMCheckpoint -VM "VM01"
PS C:\> Remove-VMCheckpoint -VMCheckpoint $Checkpoints[0] -Confirm
```

## Related topics

[Get-SCVMCheckpoint](#)

[New-SCVMCheckpoint](#)

[Restore-SCVMCheckpoint](#)

[Set-SCVMCheckpoint](#)

---

# Remove-SCVMConfiguration

---

## Remove-SCVMConfiguration

Removes a virtual machine configuration object from VMM.

### Syntax

Parameter Set: Default

```
Remove-SCVMConfiguration [-VMConfiguration] <BaseVMConfiguration> [-JobVariable <String> ]  
[-PROTipID <Guid> ] [-RunAsynchronously] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The Remove-SCVMConfiguration cmdlet removes a virtual machine configuration object from Virtual Machine Manager (VMM).

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMConfiguration<BaseVMConfiguration>**

Specifies a virtual machine configuration object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-Confirm**

Prompts you for confirmation before executing the command.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

---

|                             |       |
|-----------------------------|-------|
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Notes**

- Requires a VMM virtual machine configuration object, which can be obtained by using the Get-SCVMConfiguration cmdlet.

## **Examples**

### **1: Remove a virtual machine configuration.**

The first command gets all virtual machine configuration objects, selects the object named VMConfig01 and then stores that virtual machine configuration object in the \$VMConfig variable.

The second command removes the virtual machine configuration in \$VMConfig (VMConfig01).

```
PS C:\> $VMConfig = Get-SCVMConfiguration -All | where {$_.Name -eq "VMConfig01"}
```

```
PS C:\> Remove-SCVMConfiguration -VMConfiguration $VMConfig
```

---

## Related topics

[Get-SCVMConfiguration](#)

[New-SCVMConfiguration](#)

[Set-SCVMConfiguration](#)

[Update-SCVMConfiguration](#)

---

# Remove-SCVMHost

---

## Remove-SCVMHost

Removes a virtual machine host from VMM.

### Syntax

Parameter Set: NormalRemoval

```
Remove-SCVMHost [-VMHost] <Host> [-Credential <VMMCredential> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ForceRemoval

```
Remove-SCVMHost [-VMHost] <Host> -Force [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The Remove-SCVMHost cmdlet removes one or more virtual machine hosts from Virtual Machine Manager (VMM).

The Remove-SCVMHost cmdlet operates as follows:

- HOST SERVER ONLY - If this computer is a Hyper-V host but is not also a library server, the host object is removed from the VMM database, and the VMM agent software is uninstalled from the physical host server.

If the host is a VMware ESX host or a Citrix XenServer host, the host object is removed from the VMM database. VMM does not install an agent on ESX hosts or XenServer hosts.

- HOST AND LIBRARY SERVER - If this computer is a Hyper-V host and is also a library server, this command removes only the host functionality but leaves the library server feature in place. That is, the host object is removed from the VMM database, but the VMM agent software is not uninstalled from the physical server. The library server object remains in the database.

If the host is an ESX host, it can function only as a virtual machine host in VMM. It cannot be both a host and a library server.

- CREDENTIALS - If a Hyper-V host is joined to an Active Directory domain, you must provide credentials for an

---

account with appropriate permissions to remove that host computer from VMM.

You do not need to provide Active Directory credentials to remove a perimeter network host, an ESX host or a XenServer host from VMM.

- VIRTUAL MACHINES - When you remove a host, the host is no longer managed by VMM. However, any virtual machines on the host server will not be removed or disassociated from the server. Any running virtual machines are not shut down. Although the virtual machines are no longer managed by VMM, they are not affected in any other way.

- FORCED REMOVAL - You can use the Force parameter with the Remove-SCVMHost cmdlet to remove a virtual machine host from VMM when you do not have appropriate credentials to manage that host or when the VMM server can no longer communicate with that host.

When you specify the Force parameter, VMM will not ask or check for credentials, nor will VMM attempt to connect to the host and uninstall the VMM agent. Hence, using the Force parameter is recommended only when cleaning up stale host records from the VMM database.

This cmdlet returns the object upon success (with the property MarkedForDeletion set to TRUE) or returns an error message upon failure.

## Parameters

### **-Credential<VMMCredential>**

Specifies a credential object or, for some cmdlets, a Run As account object that contains the user name and password of an account that has permission to perform this action. Or, in the case of Restart-SCJob, has permission to complete a restarted task.

For more information about the PSCredential object, type: "Get-Help Get-Credential".

For more information about Run As accounts, type: "Get-Help New-SCRunAsAccount".

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |



---

## **-Force**

Forces the operation to complete.

For example:

- Remove-SCSCVMHost -Force

Forces the removal of a host object from the VMM database.

- Stop-SCVirtualMachine -Force

Stops a virtual machine.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | false |
| Position? | named |

|                             |       |
|-----------------------------|-------|
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VMHost<Host>**

Specifies a virtual machine host object. VMM supports Hyper-V hosts, VMware ESX hosts, and Citrix XenServer hosts.

For more information about each type of host, type: "Get-Help Add-SCVMHost -detailed". See the examples for a specific cmdlet to determine how that cmdlet uses this parameter.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-Confirm**

Prompts you for confirmation before executing the command.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                             |       |
|-----------------------------|-------|
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## -WhatIf

Describes what would happen if you executed the command without actually executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Examples

### 1: Remove a specific domain-joined host from VMM.

The first command uses the Get-Credential cmdlet to prompt you to supply a user name and password and stores the provided credentials in the \$Credential variable. The required credentials for this operation are a domain account with administrator rights to remove a Windows-based host server joined to an Active Directory domain from VMM.

The second command gets the host object named VMHost01 and stores the object in the \$VMHost variable.

The third command removes the host object stored in \$VMHost. As this command is processed, \$Credential provides credentials to Remove-VMHost, and the Confirm parameter prompts you to confirm that you do want to remove this host from VMM.

```
PS C:\> $Credential = Get-Credential
PS C:\> $VMHost = Get-SCVMHost -ComputerName "VMHost01"
PS C:\> Remove-SCVMHost -VMHost $VMHost -Credential $Credential -Confirm
```

---

## 2: Remove all hosts that are not nodes in a host cluster from VMM.

The first command gets all host objects, excludes any hosts that are nodes in a host cluster, selects only those objects that represent VMware ESX hosts, perimeter network hosts, or non-trusted domain hosts, and then removes those objects from VMM if you confirm that you want to remove them.

Credentials are not required to remove these hosts.

The second command prompts you to supply a user name and password for an account with permissions to remove domain-joined Windows hosts from VMM and stores your credentials in `$Credential`.

The last command gets all domain-joined Windows-based host objects that are not part of a host cluster and passes the objects to the `Remove-VMHost` cmdlet. As this command is processed, `$Credential` provides your credentials to `Remove-VMHost`, and the `Confirm` parameter prompts you to confirm that you do want to remove these hosts from VMM.

```
PS C:\> Get-SCVMHost | where {$_.HostCluster -eq $NULL} | where {$_.VirtualizationPlatform -eq "VMwareESX" -or $_.PerimeterNetworkHost -eq 1 -or $_.NonTrustedDomainHost -eq 1} | Remove-SCVMHost -Confirm
PS C:\> $Credential = Get-Credential
PS C:\> Get-VMHost | where {$_.HostCluster -eq $NULL -and $_.VirtualizationPlatform -ne "VMwareESX" -and $_.PerimeterNetworkHost -eq 0 -and $_.NonTrustedDomainHost -eq 0} | Remove-VMHost -Credential $Credential -Confirm
```

## 3: Remove a specific host that you can no longer access from VMM.

The first command gets the host object named `VMHost03` and stores the object in the `$VMHost` variable.

The second command switches on the `Force` parameter to ensure that `VMHost03` is removed from the VMM database. Credentials are not needed for this operation. The `Confirm` parameter prompts you to confirm that you do want to remove this host.

**NOTE:** You can use the `Force` parameter to remove a host from VMM when you do not have the credentials for that host or when the VMM server can no longer communicate with that host.

```
PS C:\> $VMHost = Get-SCVMHost -ComputerName "VMHost03"
PS C:\> Remove-SCVMHost -VMHost $VMHost -Force -Confirm
```

## Related topics

[Add-SCVMHost](#)

[Read-SCVMHost](#)

---

# Remove-SCVMHostCluster

---

## Remove-SCVMHostCluster

Removes a host cluster object from VMM.

### Syntax

Parameter Set: NormalRemoval

```
Remove-SCVMHostCluster [-VMHostCluster] <HostCluster> [-Credential <VMMCredential> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ForceRemoval

```
Remove-SCVMHostCluster [-VMHostCluster] <HostCluster> -Force [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The Remove-SCVMHostCluster cmdlet removes one or more host cluster objects from Virtual Machine Manager (VMM).

Remove-SCVMHostCluster does not destroy the cluster. To uncluster a host cluster by using VMM, use the Uninstall-SCVMHostCluster cmdlet.

### Parameters

#### **-Credential<VMMCredential>**

Specifies a credential object or, for some cmdlets, a Run As account object that contains the user name and password of an account that has permission to perform this action. Or, in the case of Restart-SCJob, has permission to complete a restarted task.

For more information about the PSCredential object, type: "Get-Help Get-Credential".

For more information about Run As accounts, type: "Get-Help New-SCRunAsAccount".

|               |       |
|---------------|-------|
| Aliases       | none  |
| Required?     | false |
| Position?     | named |
| Default Value | none  |

|                             |       |
|-----------------------------|-------|
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Force**

Forces the operation to complete.

For example:

- Remove-SCSCVMHost -Force

Forces the removal of a host object from the VMM database.

- Stop-SCVirtualMachine -Force

Stops a virtual machine.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMHostCluster<HostCluster>**

Specifies a VMM host cluster object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-Confirm**

Prompts you for confirmation before executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## -WhatIf

Describes what would happen if you executed the command without actually executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Examples

### 1: Remove a specific host cluster from VMM.

The first command gets the Run As account object named RunAsAccount01 and stores the object in the \$Credential variable. The Run As account used for this operation must be a domain account with administrator rights on all of the nodes of the failover cluster that you want to remove.

The second command gets the failover cluster object named VMHostCluster01 and stores the object in the \$VMHostCluster variable.

The last command removes the VMHostCluster01 cluster object from the VMM database and stops managing that host cluster, after prompting the user for confirmation. It does not modify the host cluster settings or its existing virtual machines. As this command is processed, \$Credential provides the stored Run As account to Remove-SCVMHostCluster.

```
PS C:\> $Credential = Get-SCRunAsAccount -Name "RunAsAccount01"
```

```
PS C:\> $Cluster = Get-SCVMHostCluster -Name "VMHostCluster01.Contoso.com"
```

```
PS C:\> Remove-SCVMHostCluster -VMHostCluster $Cluster -Credential $Credential -Confirm
```



---

## 2: Remove all host clusters from VMM.

The first command gets the Run As account object named RunAsAccount01 and stores the object in the \$Credential variable. The Run As account used for this operation must be a domain account with administrator rights on all of the nodes of the failover cluster that you want to remove.

The second command gets all host cluster objects and passes the objects to the Remove-SCVMHostCluster cmdlet, which removes each host cluster object from Virtual Machine Manager and stops managing the corresponding host cluster, after prompting the user for confirmation. The command does not modify the host cluster settings or its existing virtual machines. As this command is processed, \$Credential provides the stored Run As account to Remove-SCVMHostCluster.

```
PS C:\> $Credential = Get-SCRunAsAccount -Name "RunAsAccount01"
```

```
PS C:\> Get-SCVMHostCluster | Remove-SCVMHostCluster -Credential $Credential -Confirm
```

### Related topics

[Add-SCVMHostCluster](#)

[Find-SCCluster](#)

[Get-SCVMHostCluster](#)

[Install-SCVMHostCluster](#)

[Move-SCVMHostCluster](#)

[Read-SCVMHostCluster](#)

[Set-SCVMHostCluster](#)

[Test-SCVMHostCluster](#)

[Uninstall-SCVMHostCluster](#)

---

# Remove-SCVMHostGroup

---

## Remove-SCVMHostGroup

Removes a host group from VMM.

### Syntax

Parameter Set: Default

```
Remove-SCVMHostGroup [-VMHostGroup] <HostGroup> [-JobVariable <String> ] [-PROTipID <Guid> ]  
[-RunAsynchronously] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The Remove-SCVMHostGroup cmdlet removes one or more host group objects from the Virtual Machine Manager (VMM) database. A host group cannot be deleted if it is associated with a private cloud, if it has hosts assigned to it, or if its child host group has hosts assigned to it. This cmdlet will delete child host groups if the host group and its child host groups do not contain any virtual machine hosts.

This cmdlet returns the object upon success (with the MarkedForDeletion property set to TRUE) or returns an error message upon failure.

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VMHostGroup<HostGroup>**

Specifies a virtual machine host group object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

---

## -Confirm

Prompts you for confirmation before executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## -WhatIf

Describes what would happen if you executed the command without actually executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## Notes

- Requires a VMM host group object, which can be retrieved by using the Get-SCVMHostGroup cmdlet.

## Examples

### 1: Remove the specified host group.

The first command gets the host group named HostGroup02 and stores it in the \$HostGroup variable.

The second command removes the host group object stored in the \$HostGroup variable.

```
PS C:\> $HostGroup = Get-SCVMHostGroup -Name "HostGroup02"
```

```
PS C:\> Remove-SCVMHostGroup -VMHostGroup $HostGroup
```

---

## Related topics

[Get-SCVMHostGroup](#)

[Move-SCVMHostGroup](#)

[New-SCVMHostGroup](#)

[Set-SCVMHostGroup](#)

---

# Remove-SCVMHostNetworkAdapter

---

## Remove-SCVMHostNetworkAdapter

Removes a physical host network adapter object from a virtual network that is configured on a host managed by VMM.

### Syntax

Parameter Set: Default

```
Remove-SCVMHostNetworkAdapter [-VirtualNetwork] <VirtualNetwork> -VMHostNetworkAdapter  
<HostNetworkAdapter> [-JobGroup <Guid> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-  
RunAsynchronously] [ <CommonParameters>]
```

### Detailed Description

The Remove-SCVMHostNetworkAdapter removes one or more physical host network adapter objects from a virtual network that is configured on a host managed by Virtual Machine Manager (VMM).

### Parameters

#### **-JobGroup<Guid>**

Specifies an identifier for a series of commands that will run as a set just before the final command that includes the same job group identifier runs.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

---

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VirtualNetwork<VirtualNetwork>**

Specifies a virtual network object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-VMHostNetworkAdapter<HostNetworkAdapter>**

Specifies a physical network adapter object on a host to which virtual machines deployed on that host can connect.

Example format: -VMHostNetworkAdapter \$VMHostNIC

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Examples**

### **1: Remove the physical host network adapter from a specific virtual network.**

The first command gets the host object named VMHost01 and stores the object in the \$VMHost variable.

The second command gets the host network adapter object named HostAdapter01 and stores the object in the \$HostAdapter variable.



---

The third command gets the virtual network object named ExternalVirtualNetwork01 from VMHost01 and stores the object in the \$VirtualNetwork variable.

The last command removes the host network adapter stored in \$HostAdapter from the virtual network stored in \$VirtualNetwork. The Confirm parameter prompts you to confirm whether you want to delete the adapter from VMM.

```
PS C:\> $VMHost = Get-SCVMHost -ComputerName "VMHost01.Contoso.com"
```

```
PS C:\> $HostAdapter = Get-SCVMHostNetworkAdapter -VMHost $VMHost -Name "HostAdapter01"
```

```
PS C:\> $VirtualNetwork = Get-SCVirtualNetwork -VMHost $VMHost -Name  
"ExternalVirtualNetwork01"
```

```
PS C:\> Remove-SCVMHostNetworkAdapter -VMHostNetworkAdapter $HostAdapter -VirtualNetwork  
$VirtualNetwork -Confirm
```

## Related topics

[Add-SCVMHostNetworkAdapter](#)

[Get-SCVMHostNetworkAdapter](#)

[Set-SCVMHostNetworkAdapter](#)

---

# Remove-SCVMHostNetworkAdapterProfile

---

## Remove-SCVMHostNetworkAdapterProfile

Removes a host network adapter profile.

### Syntax

Parameter Set: Default

```
Remove-SCVMHostNetworkAdapterProfile [-VMHostNetworkAdapterProfile]  
<VMHostNetworkAdapterProfile> [-Force] [-JobVariable <String> ] [-PROTipID <Guid> ] [-  
RunAsynchronously] [ <CommonParameters>]
```

### Detailed Description

The Remove-SCVMHostNetworkAdapterProfile cmdlet removes a host network adapter profile.

### Parameters

#### -Force

Forces the operation to complete.

For example:

- Remove-SCSCVMHost -Force

Forces the removal of a host object from the VMM database.

- Stop-SCVirtualMachine -Force

Stops a virtual machine.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-VMHostNetworkAdapterProfile<VMHostNetworkAdapterProfile>**

Specifies a host network adapter profile object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Examples**

### **1. Delete a host network adapter profile.**

This command gets the host network adapter profile object with the ID of 259f47c7-c5a9-429d-a421-d232f9b34991 and then uses the pipeline operator to Remove-SCVMHostNetworkAdapterProfile which deletes the host network adapter profile.

```
PS C:\> Get-SCVMHostNetworkAdapterProfile -ID "259f47c7-c5a9-429d-a421-d232f9b34991" |  
Remove-SCVMHostNetworkAdapterProfile
```

## **Related topics**

[Get-SCVMHostNetworkAdapterProfile](#)

[New-SCVMHostNetworkAdapterProfile](#)

---

# Remove-SCVMHostProfile

---

## Remove-SCVMHostProfile

Removes a host profile object from the VMM database.

### Syntax

Parameter Set: Default

```
Remove-SCVMHostProfile [-VMHostProfile] <VMHostProfile> [-Force] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The Remove-SCVMHostProfile cmdlet removes a host profile object from the Virtual Machine Manager (VMM) database.

### Parameters

#### -Force

Forces the operation to complete.

For example:

- Remove-SCSCVMHost -Force

Forces the removal of a host object from the VMM database.

- Stop-SCVirtualMachine -Force

Stops a virtual machine.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-VMHostProfile<VMHostProfile>**

Specifies a virtual machine host profile object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-Confirm**

Prompts you for confirmation before executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Examples

### 1: Remove a specified host profile.

This command removes the host profile object named HostProfile01.

```
PS C:\> Get-SCVMHostProfile -Name "HostProfile01" | Remove-SCVMHostProfile
```

## Related topics

[Get-SCVMHostProfile](#)

[New-SCVMHostProfile](#)

[Set-SCVMHostProfile](#)



---

# Remove-SCVMNetwork

---

## Remove-SCVMNetwork

Removes a VM network.

### Syntax

Parameter Set: Default

```
Remove-SCVMNetwork [-VMNetwork] <VMNetwork> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

### Detailed Description

The Remove-SCVMNetwork cmdlet removes a VM network from Virtual Machine Manager (VMM).

To retrieve a VM network object, use the Get-SCVMNetwork cmdlet.

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-VMNetwork<VMNetwork>**

Specifies a VM network object.

To get a VM network object, use the Get-SCVMNetwork cmdlet.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-Confirm**

Prompts you for confirmation before executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

---

## Examples

### 1: Remove a VM network.

This command gets the VM network object named VMNetwork01 and uses the pipeline operator to pass the object to Remove-SCVMNetwork which deletes the VM network.

```
PS C:\> Get-SCVMNetwork -Name "VMNetwork01" | Remove-SCVMNetwork
```

## Related topics

[Get-SCVMNetwork](#)

[New-SCVMNetwork](#)

[Set-SCVMNetwork](#)

---

# Remove-SCVMNetworkGateway

---

## Remove-SCVMNetworkGateway

Removes a VM network gateway from VMM.

### Syntax

Parameter Set: Default

```
Remove-SCVMNetworkGateway [-VMNetworkGateway] <VMNetworkGateway> [-Force] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The Remove-SVVMNetworkGateway cmdlet deletes a VM network gateway from Virtual Machine Manager (VMM).

### Parameters

#### -Force

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### -JobVariable<String>

Specifies that job progress is tracked and stored in the variable named by this parameter.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

|                             |                |
|-----------------------------|----------------|
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-VMNetworkGateway<VMNetworkGateway>**

Specifies a VM network gateway object.

To obtain a VM network gateway object, use the Get-SCVMNetworkGateway cmdlet.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-Confirm**

Prompts you for confirmation before executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

---

|                             |       |
|-----------------------------|-------|
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

### Related topics

[Add-SCVMNetworkGateway](#)

[Get-SCVMNetworkGateway](#)

[Read-SCVMNetworkGateway](#)

[Set-SCVMNetworkGateway](#)



---

# Remove-SCVMSubnet

---

## Remove-SCVMSubnet

Removes a VM subnet.

### Syntax

Parameter Set: Default

```
Remove-SCVMSubnet [-VMSubnet] <VMSubnet> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

### Detailed Description

The Remove-SCVMSubnet cmdlet removes a VM subnet.

To get a VM subnet object, use the Get-SCVMSubnet cmdlet.

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-VMSubnet<VMSubnet>**

Specifies a VM subnet object.

To obtain a VM subnet object, use the Get-SCVMSubnet cmdlet.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-Confirm**

Prompts you for confirmation before executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

---

## Examples

### 1: Remove a VM subnet.

This command gets the VM subnet object named VMSubnet01 and uses the pipeline operator to pass the object to Remove-SCVMSubnet which deletes VMSubnet01.

```
PS C:\> Get-SCVMSubnet -Name "VMSubnet01" | Remove-SCVMSubnet
```

## Related topics

[Get-SCVMSubnet](#)

[New-SCVMSubnet](#)

[Set-SCVMSubnet](#)

---

# Remove-SCVMTemplate

---

## Remove-SCVMTemplate

Removes a template object from VMM and deletes all files associated with the template.

### Syntax

Parameter Set: Default

```
Remove-SCVMTemplate [-VMTemplate] <Template> [-Force] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The Remove-SCVMTemplate cmdlet removes a template object from the VMM library and deletes all files associated with the template.

The types of files that can be associated with a template include virtual hard disk files (Windows-based .vhd files, Citrix XenServer-based .vhd files, or VMware-based .vmdk files), virtual floppy disk files (Windows-based .vfd files or VMware-based .flp files), and script files (Windows PowerShell .ps1 script files or answer file scripts, including Sysprep.inf and Unattend.xml files).

This cmdlet returns the object upon success (with the property MarkedForDeletion set to TRUE) or returns an error message upon failure.

### Parameters

#### -Force

Forces the operation to complete.

For example:

- Remove-SCSCVMHost -Force

Forces the removal of a host object from the VMM database.

- Stop-SCVirtualMachine -Force

Stops a virtual machine.

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | false |
| Position? | named |

---

|                             |       |
|-----------------------------|-------|
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | false |
| Position? | named |

---

|                             |       |
|-----------------------------|-------|
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMTemplate<Template>**

Specifies a VMM template object used to create virtual machines.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-Confirm**

Prompts you for confirmation before executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Position?              | named |
| Default Value          | none  |
| Accept Pipeline Input? | false |

---

|                             |
|-----------------------------|
| Accept Wildcard Characters? |
|-----------------------------|

|       |
|-------|
| false |
|-------|

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **SCVMTemplate**

## Notes

- Requires a Virtual Machine Manager (VMM) template object, which can be retrieved by using the Get-SCVMTemplate cmdlet.

## Examples

### 1: Remove a specific template from the library.

The first command gets the template object named Template01 from the library on VMMServer01 and stores the object in the \$Template variable.

The second command removes the template object stored in \$Template from the library.

```
PS C:\> $Template = Get-SCVMTemplate -VMMServer "VMMServer1.Contoso.com" | where { $_.Name -eq "Template01" }
```

```
PS C:\> Remove-SCVMTemplate -VMTemplate $Template
```

### 2: Remove all templates from the library.

The first command gets all the template objects from VMMServer01 and stores the objects in the array named \$Templates.

The second command passes each template object in \$Templates to the Remove-SCVMTemplate cmdlet, which removes each template object from the VMM library. The Confirm parameter prompts you to confirm whether you want to delete each template.

```
PS C:\> $Templates = Get-SCVMTemplate -VMMServer "VMMServer01.Contoso.com"
```

```
PS C:\> $Templates | Remove-SCVMTemplate -Confirm
```

## Related topics

[Get-SCVMTemplate](#)



---

[New-SCVMTemplate](#)

[Set-SCVMTemplate](#)

---

# Remove-SCVMXComputerConfiguration

---

## Remove-SCVMXComputerConfiguration

Removes a VMX computer configuration object from VMM.

### Syntax

Parameter Set: Default

```
Remove-SCVMXComputerConfiguration [-VMXComputerConfiguration] <VmxMachineConfiguration> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

### Detailed Description

The Remove-SCVMXComputerConfiguration removes one or more VMX computer configuration objects from the Virtual Machine Manager (VMM) database.

This cmdlet returns the object upon success (with the property MarkedForDeletion set to TRUE) or returns an error message upon failure.

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

---

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMXComputerConfiguration<VmxMachineConfiguration>**

Specifies a VMX computer configuration object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-Confirm**

Prompts you for confirmation before executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Notes**

- Requires a VMM VMX computer configuration object, which can be retrieved by using the Get-SCVMXComputerConfiguration cmdlet.

## **Examples**

### **1: Remove all VMX computer configurations without being prompted to confirm each deletion.**

The first command retrieves all VMX computer configuration objects from the VMM database on VMMServer01 and stores these objects in the \$VMXComputerConfigs object array.

The second command passes each object in \$VMXComputerConfigs to Remove-VMXComputerConfiguration, which removes each VMX computer configuration object.

```
PS C:\> $VMXComputerConfigs = Get-SCVMXComputerConfiguration -VMMServer
"VMMServer01.Contoso.com"
```

---

PS C:\> \$VMXComputerConfigs | Remove-SCVMXComputerConfiguration

## Related topics

[Get-SCVMXComputerConfiguration](#)

[New-SCV2V](#)

[New-SCVMXComputerConfiguration](#)

---

# Remove-SCVPNConnection

---

## Remove-SCVPNConnection

Removes a VPN connection from VMM.

### Syntax

Parameter Set: Default

```
Remove-SCVPNConnection [-VPNConnection] <VPNConnection> [-Force] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The Remove-SCVPNConnection cmdlet deletes a virtual private network (VPN) connection from Virtual Machine Manager (VMM).

### Parameters

#### -Force

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### -JobVariable<String>

Specifies that job progress is tracked and stored in the variable named by this parameter.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

|                             |                |
|-----------------------------|----------------|
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-VPNConnection<VPNConnection>**

Specifies a virtual private network (VPN) connection object.

To obtain a VPN connection object, use the Get-SCVPNConnection cmdlet.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-Confirm**

Prompts you for confirmation before executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|



---

|                             |       |
|-----------------------------|-------|
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

### Related topics

[Add-SCVPNConnection](#)

[Get-SCVPNConnection](#)

[Set-SCVPNConnection](#)

---

# Repair-SCStorageFileShare

---

## Repair-SCStorageFileShare

Repairs a storage file share.

### Syntax

Parameter Set: RepairShareWithHost

```
Repair-SCStorageFileShare -StorageFileShare <StorageFileShare> -VMHost <Host> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

Parameter Set: RepairShareWithCluster

```
Repair-SCStorageFileShare -StorageFileShare <StorageFileShare> -VMHostCluster <HostCluster> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

Parameter Set: RepairShareWithLibrary

```
Repair-SCStorageFileShare -LibraryServer <LibraryServer> -StorageFileShare <StorageFileShare> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

### Detailed Description

The Repair-SCStorageFileShare cmdlet repairs a specified storage file share.

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-LibraryServer<LibraryServer>**

Specifies a VMM library server object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-StorageFileShare<StorageFileShare>**

Specifies a storage file share.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VMHost<Host>**

Specifies a virtual machine host object. VMM supports Hyper-V hosts, VMware ESX hosts, and Citrix XenServer hosts.

For more information about each type of host, type: "Get-Help Add-SCVMHost -detailed". See the examples for a specific cmdlet to determine how that cmdlet uses this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VMHostCluster<HostCluster>**

Specifies a VMM host cluster object.

|                        |       |
|------------------------|-------|
| Aliases                | none  |
| Required?              | true  |
| Position?              | named |
| Default Value          | none  |
| Accept Pipeline Input? | false |

---

|                             |
|-----------------------------|
| Accept Wildcard Characters? |
|-----------------------------|

|       |
|-------|
| false |
|-------|

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Examples

### 1: Repair a storage file share by providing a virtual machine host.

The first command gets the storage file share object named FileShare01 and stores the object in the \$FileShare variable.

The second command repairs FileShare01.

```
PS C:\> $FileShare = Get-SCStorageFileShare -Name "FileShare01"
```

```
PS C:\> $VMHost = Get-SCVMHost -Computername "VMHost01"
```

```
PS C:\> Repair-SCStorageFileShare -StorageFileShare $FileShare -VMHost $VMHost
```

## Related topics

[Get-SCStorageFileShare](#)

[New-SCStorageFileShare](#)

[Register-SCStorageFileShare](#)

[Remove-SCStorageFileShare](#)

[Set-SCStorageFileShare](#)

[Unregister-SCStorageFileShare](#)

---

# Repair-SCVirtualMachine

---

## Repair-SCVirtualMachine

Repairs a virtual machine in a failed state.

### Syntax

Parameter Set: Agent

```
Repair-SCVirtualMachine [-VM] <VM> -Agent [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

Parameter Set: Dismiss

```
Repair-SCVirtualMachine [-VM] <VM> -Dismiss [-Force] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

Parameter Set: Retry

```
Repair-SCVirtualMachine [-VM] <VM> -Retry [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

Parameter Set: Undo

```
Repair-SCVirtualMachine [-VM] <VM> -Undo [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

### Detailed Description

The Repair-SCVirtualMachine cmdlet repairs a virtual machine in a failed state that is on a host managed by Virtual Machine Manager (VMM). A virtual machine can be in one of four types of failed state:

- Creation Failed
- Migration Failed
- Update Failed
- Deletion Failed

You can use this command to repair a failure as follows:

- \* **RETRY**. Attempts to perform the failed job again.
- \* **UNDO**. Attempts to undo any changes made to the virtual machine and restore it to a healthy state. For example, if a Move-SCVirtualMachine job fails, using the Undo option attempts to move the virtual machine back to its previous host.

---

\* DISMISS. Dismisses the failed job and refreshes the virtual machine based on its current state. If you manually fix a failure (for example, by manually moving the .vhd and .vmc files to a new host after a failed Move-SCVirtualMachine attempt), you can use the Dismiss option to refresh the data for the virtual machine in the VMM database. However, using the Dismiss option might return the object to the failed state.

When you run Repair-SCVirtualMachine, you can specify only one type of action at a time.

You can run Repair-SCVirtualMachine to repair an in-guest agent for a virtual machine that is part of a service by using the Agent parameter.

## Parameters

### -Agent

Indicates that the VMM in-guest agent should be repaired or upgraded, as applicable.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### -Dismiss

Dismisses the error on an object or an update notification on a service instance.

After an error is dismissed, the object is refreshed. If the error reappears, refreshing does not solve the problem and you must fix the error.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-Force**

Forces the operation to complete.

For example:

- Remove-SCSCVMHost -Force

Forces the removal of a host object from the VMM database.

- Stop-SCVirtualMachine -Force

Stops a virtual machine.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | false |



---

|                             |       |
|-----------------------------|-------|
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Retry**

Retries the last task that failed on a VMM object in an attempt to complete the task successfully.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Undo**

Cancels the last job run on a VMM object and reverses any changes that were made. This parameter is available only if the most recent job failed.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

|                             |       |
|-----------------------------|-------|
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VM<VM>**

Specifies a virtual machine object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **VirtualMachine**

## **Notes**

- Requires a virtual machine object, which can be retrieved by using the Get-SCVirtual Machine cmdlet.

---

## Examples

### 1: Repair a failed migration task by retrying the migration task.

The first command gets the virtual machine object named VM01 and stores the object in the \$VM variable. This example assumes that the task that you want to repair by using the Retry parameter is an attempt to move (migrate) the virtual machine from one host to another.

The second command repairs the virtual machine object stored in \$VM (in this case, VM01) by restarting the previous failed migration task.

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM01"
```

```
PS C:\> Repair-SCVirtualMachine -VM $VM -Retry
```

### 2: Repair or upgrade the VMM In-guest agent on a specified virtual machine that is part of a service.

The first command gets the virtual machine object named ServiceVM01 and stores the object in the \$VM variable. This example assumes that the virtual machine is part of a service.

The second command gets a credential object, which must be a local administrator on the virtual machine to be repaired and stores the object in the \$Creds variable

The third command repairs the in-guest agent on the virtual machine object stored in \$VM (ServiceVM01).

```
PS C:\> $VM = Get-SCVirtualMachine -Name "ServiceVM01"
```

```
PS C:\> $Creds = Get-Credential
```

```
PS C:\> Repair-SCVirtualMachine -VM $VM -Credential $Creds -Agent
```

## Related topics

[Get-SCVirtualMachine](#)

[Move-SCVirtualMachine](#)

[New-SCVirtualMachine](#)

[Read-SCVirtualMachine](#)

[Register-SCVirtualMachine](#)

[Remove-SCVirtualMachine](#)

[Reset-SCVirtualMachine](#)

[Resume-SCVirtualMachine](#)

[Set-SCVirtualMachine](#)

[Start-SCVirtualMachine](#)

[Stop-SCVirtualMachine](#)

[Suspend-SCVirtualMachine](#)

---

# Repair-SCVirtualNetwork

---

## Repair-SCVirtualNetwork

Repairs a virtual network.

### Syntax

Parameter Set: Default

```
Repair-SCVirtualNetwork [-VirtualNetwork] <VirtualNetwork> [-JobGroup <Guid> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

### Detailed Description

The Repair-SCVirtualNetwork cmdlet repairs a virtual network.

### Parameters

#### **-JobGroup<Guid>**

Specifies an identifier for a series of commands that will run as a set just before the final command that includes the same job group identifier runs.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

---

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VirtualNetwork<VirtualNetwork>**

Specifies a virtual network object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-Confirm**

Prompts you for confirmation before executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

---

## Examples

### 1: Repair a virtual network.

```
PS C:\> $VirtNetwork = Get-SCVirtualNetwork -Name "VirtualNetwork01"
```

```
PS C:\> Repair-SCVirtualNetwork -VirtualNetwork $VirtNetwork
```

## Related topics

[Get-SCVirtualNetwork](#)

[New-SCVirtualNetwork](#)

[Remove-SCVirtualNetwork](#)

[Set-SCVirtualNetwork](#)

---

# Repair-SCVirtualNetworkAdapter

---

## Repair-SCVirtualNetworkAdapter

Repairs a virtual network adapter.

### Syntax

Parameter Set: Default

```
Repair-SCVirtualNetworkAdapter [-VirtualNetworkAdapter] <VirtualNetworkAdapter> [-JobGroup <Guid> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Repair-SCVirtualNetworkAdapter cmdlet repairs a virtual network adapter.

### Parameters

#### **-JobGroup<Guid>**

Specifies an identifier for a series of commands that will run as a set just before the final command that includes the same job group identifier runs.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.



|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VirtualNetworkAdapter<VirtualNetworkAdapter>**

Specifies a virtual network adapter object for a virtual machine.

TYPE OF HOST      NUMBER OF VIRTUAL NETWORK ADAPTERS

---

-----

Hyper-V Up to 4 emulated adapters per virtual machine.  
 Up to 8 synthetic adapters per virtual machine.  
 (Exception: no driver available for an emulated network adapter on a Windows Server 2003 x64 guest.)

VMware ESX Up to 4 emulated adapters per virtual machine.

Citrix XenServer Up to 7 emulated adapters per virtual machine.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

**-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

**<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

---

## Examples

### 1: Repair a virtual network adapter.

The first command gets the virtual machine object named VM01 and stores the object in the \$VM variable.

The second command gets all virtual network adapter objects for VM01 and stores the objects in the \$VirtNetworkAdapter variable.

The last command repairs the first virtual network adapter stored in \$VirtNetworkAdapter.

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM01"
```

```
PS C:\> $VirtNetworkAdapter = Get-SCVirtualNetworkAdapter -VM $VM
```

```
PS C:\> Repair-SCVirtualNetworkAdapter -VirtualNetworkAdapter $VirtNetworkAdapter[0]
```

### Related topics

[Get-SCVirtualNetworkAdapter](#)

[New-SCVirtualNetworkAdapter](#)

[Remove-SCVirtualNetworkAdapter](#)

[Set-SCVirtualNetworkAdapter](#)

---

# Repair-SCVMHost

---

## Repair-SCVMHost

Starts remediation steps on a failed host for a set of known failure conditions.

### Syntax

Parameter Set: Default

```
Repair-SCVMHost [-VMHost] <Host> [-Credential <VMMCredential> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

### Detailed Description

The Repair-SCVMHost cmdlet starts a set of remediation steps on a host in failed state for a set of known failure conditions.

### Parameters

#### **-Credential<VMMCredential>**

Specifies a credential object or, for some cmdlets, a Run As account object that contains the user name and password of an account that has permission to perform this action. Or, in the case of Restart-SCJob, has permission to complete a restarted task.

For more information about the PSCredential object, type: "Get-Help Get-Credential".

For more information about Run As accounts, type: "Get-Help New-SCRunAsAccount".

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

---

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-VMHost<Host>**

Specifies a virtual machine host object. VMM supports Hyper-V hosts, VMware ESX hosts, and Citrix XenServer hosts.

For more information about each type of host, type: "Get-Help Add-SCVMHost -detailed". See the examples for a specific cmdlet to determine how that cmdlet uses this parameter.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **Host**
- **Host[]**

## **Examples**

### **1: Start remediation steps for a failed host object.**

The first command gets the Run As account object named RunAsAccount01 and stores the object in the \$RunAsAccount variable.

The second command gets the host object named VMHost01 and uses the pipeline operator to pass the object to the Repair-SCVMHost cmdlet which triggers the remediation steps for known failure causes, using the credentials supplied in \$RunAsAccount.

```
PS C:\> $RunAsAccount = Get-SCRunAsAccount -Name "RunAsAccount01"
```

```
PS C:\> Get-SCVMHost -ComputerName "VMHost01" | Repair-SCVMHost -Credential $RunAsAccount
```

## **Related topics**

[Add-SCVMHost](#)

---

[Disable-SCVMHost](#)

[Enable-SCVMHost](#)

[Get-SCVMHost](#)

[Move-SCVMHost](#)

[New-SCVMHost](#)

[Read-SCVMHost](#)

[Register-SCVMHost](#)

[Remove-SCVMHost](#)

[Restart-SCVMHost](#)

[Set-SCVMHost](#)

[Start-SCVMHost](#)

[Stop-SCVMHost](#)

---

# Reset-SCPROMonitorState

---

## Reset-SCPROMonitorState

Resets the state of a specified PRO monitor.

### Syntax

Parameter Set: Default

```
Reset-SCPROMonitorState -PROMonitorState <PROMonitorState> [-VMMServer <ServerConnection> ]  
[ <CommonParameters>]
```

### Detailed Description

The Reset-SCPROMonitorState cmdlet resets the state of a specified PRO Monitor.

### Parameters

#### **-PROMonitorState<PROMonitorState>**

Specifies a PRO monitor state object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

#### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|



---

|                             |                |
|-----------------------------|----------------|
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **PROMonitorState**

## Examples

### 1: Reset a PRO monitor state.

The first command gets the host object named VMHost01 and stores the object in the \$VMHost variable.

The second command gets all PRO monitor state objects on VMHost01 and stores the objects in the \$PROMonitorState object array. This example assumes that there are multiple PRO monitors on VMHost01.

The last command resets the first PRO monitor state object stored in \$PROMonitorState.

```
PS C:\> $VMHost = Get-SCVMHost "VMHost01.Contoso.com"
PS C:\> $PROMonitorState = @(Get-SCPROMonitorState -VMHost $VMHost)
PS C:\> Reset-SCProMonitorState -PROMonitorState $PROMonitorState[0]
```

## Related topics

[Get-SCPROMonitorState](#)

---

# Reset-SCVirtualMachine

---

## Reset-SCVirtualMachine

Resets a virtual machine by powering off the virtual machine and then starting it.

### Syntax

Parameter Set: SingleVM

```
Reset-SCVirtualMachine [-VM] <VM> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

### Detailed Description

The Reset-SCVirtualMachine cmdlet resets a virtual machine by powering off the virtual machine and then starting it. This is the equivalent of using the Stop-SCVirtualMachine and Start-SCVirtualMachine cmdlets. To use Reset-SCVirtualMachine, the virtual machine needs to be in a running state.

For information about stopping a virtual machine, type: "Get-Help Stop-SCVirtualMachine -detailed".

For information about starting a virtual machine, type: "Get-Help Start-SCVirtualMachine -detailed".

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

---

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VM<VM>**

Specifies a virtual machine object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **VirtualMachine**

## Notes

- Requires a VMM virtual machine object, which can be retrieved by using the Get-SCVirtualMachine cmdlet.

## Examples

### 1: Reset a virtual machine.

The first command gets the virtual machine object named VM01 and stores the object in the \$VM variable.

The second command resets the virtual machine stored in \$VM.

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM01"
```

```
PS C:\> Reset-SCVirtualMachine -VM $VM
```

## Related topics

[Get-SCVirtualMachine](#)

[Move-SCVirtualMachine](#)

[New-SCVirtualMachine](#)

[Read-SCVirtualMachine](#)

[Register-SCVirtualMachine](#)

[Remove-SCVirtualMachine](#)

[Repair-SCVirtualMachine](#)

[Resume-SCVirtualMachine](#)

[Save-SCVirtualMachine](#)

[Set-SCVirtualMachine](#)

[Start-SCVirtualMachine](#)

[Stop-SCVirtualMachine](#)

[Suspend-SCVirtualMachine](#)

---

# Resolve-SCServiceTemplate

---

## Resolve-SCServiceTemplate

Validates a service template and updates the global settings for the service template.

### Syntax

Parameter Set: Default

```
Resolve-SCServiceTemplate [-ServiceTemplate] <ServiceTemplate> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-Update] [ <CommonParameters>]
```

### Detailed Description

The Resolve-SCServiceTemplate cmdlet validates a service template and updates the global settings for the service template.

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-ServiceTemplate<ServiceTemplate>**

Specifies a service template object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-Update**

Updates the settings for an object.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

---

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **ServiceTemplate**

## Examples

### 1: Validate the global settings in a service template.

The first command gets the Beta release of the service template object named ServiceTemplate01 and stores the object in the \$SvcTemplate variable.

The second command validates the global settings for the service template in \$SvcTemplate and then displays the warnings (if any) for changes to the global settings.

```
PS C:\> $SvcTemplate = Get-SCServiceTemplate -Name "ServiceTemplate01" | where { $_.Release -eq "Beta" }
```

```
PS C:\> Resolve-SCServiceTemplate -ServiceTemplate $SvcTemplate
```

### 2: Update the global settings in a specific service template.

The first gets the service template object that named ServiceTemplate01 with a release value of Beta and stores the object in the \$SvcTemplate variable.

The second command updates the global settings for the service template stored in \$SvcTemplate.

```
PS C:\> $SvcTemplate = Get-SCServiceTemplate -Name "ServiceTemplate01" | where { $_.Release -eq "Beta" }
```

```
PS C:\> Resolve-SCServiceTemplate -ServiceTemplate $SvcTemplate -Update
```

---

## Related topics

[Get-SCServiceTemplate](#)

[New-SCServiceTemplate](#)

[Read-SCServiceTemplate](#)

[Remove-SCServiceTemplate](#)

[Set-SCServiceTemplate](#)

[Test-SCServiceTemplate](#)



---

# Restart-SCJob

---

## Restart-SCJob

Restarts a failed or canceled VMM job.

### Syntax

Parameter Set: Default

```
Restart-SCJob [-Job] <Task> [-Credential <VMMCredential> ] [-SkipLastFailedStep] [  
<CommonParameters>]
```

### Detailed Description

The Restart-SCJob cmdlet restarts one or more Virtual Machine Manager (VMM) jobs that have failed or that have been canceled by a user. Jobs that are currently running must be canceled before they can be restarted. All restarted jobs start from the last known good checkpoint before a failure or a cancellation (some jobs have only a single checkpoint).

Restarting a job displays the object properties of the job to the user and shows the "Status" property as "Running".

### Parameters

#### **-Credential<VMMCredential>**

Specifies a credential object or, for some cmdlets, a Run As account object that contains the user name and password of an account that has permission to perform this action. Or, in the case of Restart-SCJob, has permission to complete a restarted task.

For more information about the PSCredential object, type: "Get-Help Get-Credential".

For more information about Run As accounts, type: "Get-Help New-SCRunAsAccount".

|                        |                |
|------------------------|----------------|
| Aliases                | none           |
| Required?              | false          |
| Position?              | named          |
| Default Value          | none           |
| Accept Pipeline Input? | True (ByValue) |

---

|                             |       |
|-----------------------------|-------|
| Accept Wildcard Characters? | false |
|-----------------------------|-------|

### **-Job<Task>**

Specifies a VMM job object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-SkipLastFailedStep**

Indicates that the last step that failed will not be rerun when a job is restarted.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

### **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **Job**

---

## Notes

- Requires a VMM job, which can be retrieved by using the Get-SCJob cmdlet.

## Examples

### 1: Restart all jobs that were cancelled on a specific virtual machine.

The first command gets the Run As account object named RunAsAccount01 and stores the object in the \$Creds variable.

The second command gets all VMM jobs from the VMM database, selects only jobs on virtual machine VM01 that have been cancelled, and then passes each object to the Restart-Job cmdlet, which restarts the jobs using the Run As account supplied in \$Creds.

```
PS C:\> $Creds = Get-SCRunAsAccount -Name "RunAsAccount01"
PS C:\> Get-SCJob | where { $_.ResultName -eq "VM01" -and $_.Status -eq "Canceled" } |
Restart-SCJob -Credential $Creds
```

### 2: Restart a specific job.

The first command gets the Run As account object named RunAsAccount01 and stores the object in the \$Creds variable.

The second command gets the VMM job object with ID cb3a0f0a-9fbc-4bd0-a999-3fae8cd77177, and restarts that job using the Run As account supplied in \$Creds.

```
PS C:\> $Creds = Get-SCRunAsAccount -Name "RunAsAccount01"
PS C:\> Get-SCJob -ID "cb3a0f0a-9fbc-4bd0-a999-3fae8cd77177" | Restart-SCJob -Credential
$Creds
```

## Related topics

[Find-SCJob](#)

[Get-SCJob](#)

[Stop-SCJob](#)

---

# Restart-SCVMHost

---

## Restart-SCVMHost

Restarts a specified host managed by VMM.

### Syntax

Parameter Set: Default

```
Restart-SCVMHost [-VMHost] <Host> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-UseOutOfBandChannel] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The Restart-SCVMHost cmdlet restarts a specified host managed by Virtual Machine Manager (VMM). Restart-SCVMHost restarts the host by issuing "Power Off" and "Power On" commands.

There are two methods by which you can restart a host: in-band and out-of-band. The default parameter set uses the in-band method. To use the out-of-band method, the host must be configured for out-of-band (OOB) management. For information about configuring the Baseboard Management Controller (BMC) settings for a host, type: "Get-Help Set-SCVMHost -detailed".

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-UseOutOfBandChannel**

Indicates that the out-of-band management channel should be used to perform the operation. This parameter should be used only when the computer is not responsive.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-VMHost<Host>**

Specifies a virtual machine host object. VMM supports Hyper-V hosts, VMware ESX hosts, and Citrix XenServer hosts.

For more information about each type of host, type: "Get-Help Add-SCVMHost -detailed". See the examples for a specific cmdlet to determine how that cmdlet uses this parameter.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-Confirm**

Prompts you for confirmation before executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- Host
- Host[]

## Examples

### 1: Restart a specified host.

This command restarts the host named NewHost01 using in-band management.

```
PS C:\> Get-SCVMHost -ComputerName "NewHost01" | Restart-SCVMHost
```

### 2: Restart a specified host using OOB management.

This command restarts the host named NewHost01 using out-of-band (OOB) management. The command prompts you to confirm the action before proceeding.

Note: The host must be configured for OOB management prior to running this command.

```
PS C:\> Get-SCVMHost -ComputerName "NewHost01" | Restart-SCVMHost -UseOutOfBandChannel -Confirm
```

## Related topics

[Add-SCVMHost](#)

[New-SCVMHost](#)

[Set-SCVMHost](#)

[Start-SCVMHost](#)

[Stop-SCVMHost](#)

---

# Restore-SCVMCheckpoint

---

## Restore-SCVMCheckpoint

Restores a virtual machine to a specified checkpoint.

### Syntax

Parameter Set: Default

```
Restore-SCVMCheckpoint -VMCheckpoint <VMCheckpoint> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

### Detailed Description

The Restore-SCVMCheckpoint restores a virtual machine to a specified checkpoint. A virtual machine checkpoint is a point-in-time "snapshot" of a virtual machine. You can use the checkpoint to revert a virtual machine to a previous state.

If the restore operation is successful, the Restore-VMCheckpoint cmdlet returns (displays) the checkpoint object. If the operation fails, the cmdlet returns an error message.

Restoring a virtual machine to an earlier checkpoint discards all changes made to the virtual machine since the most recent checkpoint was created. However, all checkpoints, including those made after the checkpoint to which you restore a virtual machine, remain available. Therefore, a good practice is to create a new checkpoint before you restore the virtual machine to ensure that the current state of the virtual machine is available after the restore operation.

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |



---

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VMCheckpoint<VMCheckpoint>**

Specifies a VMM virtual machine checkpoint object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **VMCheckpoint**

## Notes

- Requires a VMM checkpoint object, which can be retrieved by using the Get-SCVMCheckpoint cmdlet.

## Examples

### 1: Restore a virtual machine to its most recent checkpoint.

This command gets the most recent checkpoint object for virtual machine VM01 and restores VM01 to the state that it was in at the time its most recent checkpoint was created.

```
PS C:\> Get-SCVMCheckpoint -VM "VM01" -MostRecent | Restore-SCVMCheckpoint
```

### 2: Restore a virtual machine to the specified checkpoint.

The first command gets all checkpoint objects for virtual machine VM02 and stores the objects in the \$Checkpoints object array.

The second command restores VM02 to the second-from-last checkpoint (this example assumes you have at least 2 checkpoints). VMM retains the checkpoints created after the checkpoint that you restore to, enabling you to restore the virtual machine to a later checkpoint. To restore a virtual machine to its most recent checkpoint, see Example 1.

```
PS C:\> $Checkpoints = Get-VMCheckpoint -VM "VM02"
```

```
PS C:\> Restore-VMCheckpoint -VMCheckpoint $Checkpoints[$Checkpoints.count - 2]
```

### 3: View the hardware profile of the last restored checkpoint on a virtual machine.

The first command gets the virtual machine object named VM02 and stores the object in the \$VM variable. This example assumes that the virtual machine has been restored to one of its checkpoints.

The second command displays information about the hardware profile of the last restored checkpoint on VM02.

---

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM02"
```

```
PS C:\> $VM.LastRestoredVMCheckpoint.CheckpointHWProfile
```

## Related topics

[Get-SCVMCheckpoint](#)

[New-SCVMCheckpoint](#)

[Remove-SCVMCheckpoint](#)

[Set-SCVMCheckpoint](#)

---

# Resume-SCService

---

## Resume-SCService

Starts a paused VMM service and all virtual machines within that service.

### Syntax

Parameter Set: Default

```
Resume-SCService [-Service] <Service> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Resume-SCService cmdlet starts a paused Virtual Machine Manager (VMM) service and all virtual machines within that service. To pause a service use the Suspend-SCService cmdlet.

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Service<Service>**

Specifies a VMM service object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

---

|                             |                |
|-----------------------------|----------------|
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **VirtualMachine[]**

## Examples

### 1: Resume a specific service.

The first command gets the service object named Service01 and stores the object in the \$Service variable.

The second command resumes the service stored in \$Service, which resumes all of the virtual machines in the service.

```
PS C:\> $Service = Get-SCService -Name "Service01"
```

```
PS C:\> Resume-SCService -Service $Service
```

## Related topics

[Get-SCService](#)

[New-SCService](#)

[Read-SCService](#)

[Remove-SCService](#)

[Set-SCService](#)

[Start-SCService](#)

[Stop-SCService](#)

[Suspend-SCService](#)

[Update-SCService](#)

---

# Resume-SCVirtualMachine

---

## Resume-SCVirtualMachine

Resumes paused virtual machines managed by VMM.

### Syntax

Parameter Set: Default

```
Resume-SCVirtualMachine [-VM] <VM> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

### Detailed Description

The Resume-SCVirtualMachine cmdlet resumes one or more paused virtual machines managed by Virtual Machine Manager (VMM). A paused virtual machine is one that has been suspended by using the Suspend-SCVirtualMachine cmdlet. Using the Resume-SCVirtualMachine cmdlet to resume a virtual machine returns its object in a Running state. When the virtual machine is running again, the user can resume activity on that virtual machine.

If you run Resume-SCVirtualMachine on a virtual machine that is already running, the cmdlet returns an error message indicating that the virtual machine is not in a state that Resume-SCVirtualMachine can act on.

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VM<VM>**

Specifies a virtual machine object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |



---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **VirtualMachine**

## Notes

- Requires a virtual machine object, which can be retrieved by using the Get-SCVirtualMachine cmdlet.

## Examples

### 1: Resume a paused virtual machine.

The first command gets the virtual machine object named VM01 (which this example assumes to be in a paused state) and stores the object in the \$VM variable.

The second command resumes the virtual machine stored in \$VM (in this case, VM01) to a running state and displays information about the object to the user.

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM01"
```

```
PS C:\> Resume-SCVirtualMachine -VM $VM
```

### 2: Resume all paused virtual machines.

The first command gets all virtual machine objects from VMMServer01 that are paused, and then stores the objects in the \$VMs object array.

The second command passes each object stored in \$VMs to Resume-VM, which resumes each virtual machine to a running state.

```
PS C:\> $VMs = Get-SCVirtualMachine -VMMServer "VMMServer01.Contoso.com" | where {$_.Status -eq "Paused"}
```

```
PS C:\> $VMs | Resume-SCVirtualMachine
```

## Related topics

[Get-SCVirtualMachine](#)

[Move-SCVirtualMachine](#)

[New-SCVirtualMachine](#)

[Read-SCVirtualMachine](#)

---

[Register-SCVirtualMachine](#)

[Remove-SCVirtualMachine](#)

[Repair-SCVirtualMachine](#)

[Reset-SCVirtualMachine](#)

[Set-SCVirtualMachine](#)

[Start-SCVirtualMachine](#)

[Stop-SCVirtualMachine](#)

[Suspend-SCVirtualMachine](#)

---

# Revoke-SCIPAddress

---

## Revoke-SCIPAddress

Returns an allocated IP address to the static IP address pool.

### Syntax

Parameter Set: Default

```
Revoke-SCIPAddress [-AllocatedIPAddress] <AllocatedIPAddress> [-JobVariable <String> ] [-  
PROTipID <Guid> ] [-ReturnToPool <Boolean> ] [-RunAsynchronously] [-VMMServer  
<ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Revoke-SCIPAddress cmdlet returns an allocated IP address to the static IP address pool.

For information about allocating IP addresses, type: "Get-Help Grant-SCIPAddress -detailed".

### Parameters

#### **-AllocatedIPAddress<AllocatedIPAddress>**

Specifies an IP address that has been allocated from an IP address pool.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-ReturnToPool<Boolean>**

Indicates whether an IP address or MAC address is returned to its address pool. By default, this value is set to True.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **CloudDRPairingInfoData**

## **Examples**

### **1: Return an unassigned allocated IP address to the IP address pool.**

The first command gets the static IP address pool object with the IPv4 subnet of 10.0.0.0/24 and stores the object in the \$IPAddressPool variable.

The second command gets all unassigned allocated IP address objects for the static IP address pool stored in \$IPAddressPool and stores the objects in the \$IPAddress variable.

---

The last command revokes the first IP address stored in \$IPAddress and returns the address to the IP address pool.

```
PS C:\> $IPAddressPool = Get-SCStaticIPAddressPool -IPv4 -Subnet "10.0.0.0/24"
```

```
PS C:\> $IPAddress = Get-SCIPAddress -StaticIPAddressPool $IPAddressPool -Unassigned
```

```
PS C:\> Revoke-SCIPAddress -AllocatedIPAddress $IPAddress[0]
```

## Related topics

[Get-SCIPAddress](#)

[Grant-SCIPAddress](#)

[Set-SCIPAddress](#)

---

# Revoke-SCMACAddress

---

## Revoke-SCMACAddress

Returns an allocated MAC address to the MAC address pool.

### Syntax

Parameter Set: Default

```
Revoke-SCMACAddress [-AllocatedMACAddress] <AllocatedMACAddress> [-JobVariable <String> ] [-  
PROTipID <Guid> ] [-ReturnToPool <Boolean> ] [-RunAsynchronously] [-VMMServer  
<ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Revoke-SCMACAddress cmdlet returns an allocated MAC address to the MAC address pool. For information about granting MAC addresses, type: "Get-Help Grant-SCMACAddress -detailed".

### Parameters

#### **-AllocatedMACAddress<AllocatedMACAddress>**

Specifies a Media Access Control (MAC) address that has been allocated from a MAC address pool.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-ReturnToPool<Boolean>**

Indicates whether an IP address or MAC address is returned to its address pool. By default, this value is set to True.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.



|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **CloudPairing[]**

## **Examples**

### **1: Return an allocated MAC address to the MAC address pool.**

The first command gets the host group object at path "All Hosts\HostGroup02\Production" and stores the object in the \$HostGroup variable.

The second command gets the MAC address pools for the host group stored in \$HostGroup and stores the objects in the \$MACAddressPool array.

---

The third command gets the allocated MAC addresses from the first MAC address pool stored in \$MACAddressPool and stores the objects in \$MACAddress.

The last command revokes the first MAC address stored in \$MACAddress.

```
PS C:\> $HostGroup = Get-SCVMHostGroup | where { $_.Path -eq "All  
Hosts\HostGroup02\Production" }
```

```
PS C:\> $MACAddressPool = Get-SCMACAddressPool -VMHostGroup $HostGroup
```

```
PS C:\> $MACAddress = Get-SCMACAddress -MACAddressPool $MACAddressPool[0]
```

```
PS C:\> Revoke-SCMACAddress $MACAddress[0]
```

## Related topics

[Get-SCMACAddress](#)

[Get-SCMACAddressPool](#)

[Grant-SCMACAddress](#)

---

# Revoke-SCResource

---

## Revoke-SCResource

Revokes access to a resource from a user or user role.

### Syntax

Parameter Set: Default

```
Revoke-SCResource -Resource <ClientObject> [-JobGroup <Guid> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-UserName <String> ] [-UserRoleID <Guid[]> ] [-UserRoleName <String[]> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Revoke-SCResource cmdlet revokes access to a resource from a user or user role.

### Parameters

#### **-JobGroup<Guid>**

Specifies an identifier for a series of commands that will run as a set just before the final command that includes the same job group identifier runs.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Resource<ClientObject>**

Specifies a resource object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-UserName<String>**

Specifies a the name of a user. Enter a user name with the format Domain\User.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-UserRoleID<Guid[]>**

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-UserRoleName<String[]>**

Specifies the name of a user role. Types of user roles that are named include Delegated Administrator, Read-Only Administrator and Self-Service User.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **Resource**

## **Examples**

### **1: Revoke access to a resource from a specific user.**

The first command gets the template object named Template01 and stores the object in the \$Resource variable.

The second command revokes access to the resource stored in \$Resource (Template01) from the user named Katarina. If the user is a member of multiple user roles, access will be revoked from the user in all it's user roles.

---

```
PS C:\> $Resource = Get-SCVMTemplate | where {$_.Name -eq "Template01"}
PS C:\> Revoke-SCResource -Resource $Resource -Username "Contoso\Katarina"
```

## 2: Revoke access to a resource from a user who is a member of multiple user roles.

The first command gets the template object named Template01 and stores the object in the \$Resource variable.

The second command revokes access to the resource stored in \$Resource (Template01) from the user named Katarina, but only if the user is using the ContosoSelfServiceUsers or SelfServiceUserRole02 user roles. If Katarina uses a different user role that has access to the resource then she will still be able to access the resource.

```
PS C:\> $Resource = Get-SCVMTemplate | where {$_.Name -eq "Template01"}
PS C:\> Revoke-SCResource -Resource $Resource -Username "Contoso\Katarina" -UserRoleName
@("ContosoSelfServiceUsers", "SelfServiceUserRole02")
```

## 3: Revoke access to a resource from all members of a user role.

The first command gets the template object named Template01 and stores the object in the \$Resource variable.

The second command revokes access to the resource stored in \$Resource (Template01) from all members of the ContosoSelfServiceUsers user role.

```
PS C:\> $Resource = Get-SCVMTemplate | where {$_.Name -eq "Template01"}
PS C:\> Revoke-SCResource -Resource $Resource -UserRoleName "ContosoSelfServiceUsers"
```

## Related topics

[Grant-SCResource](#)

---

# Save-SCVirtualMachine

---

## Save-SCVirtualMachine

Migrates a virtual machine deployed on a host to the VMM library.

### Syntax

Parameter Set: Default

```
Save-SCVirtualMachine [-VM] <VM> [-JobVariable <String> ] [-LibraryServer <LibraryServer> ]  
[-PROTipID <Guid> ] [-RunAsynchronously] [-SharePath <String> ] [-UseLAN] [  
<CommonParameters>]
```

### Detailed Description

The Save-SCVirtualMachine cmdlet migrates a virtual machine deployed on a host to the VMM library. The Save-SCVirtualMachine cmdlet lets you store a virtual machine to the VMM library by using one of the following transfer methods:

- SAN TRANSFER (Fibre Channel, iSCSI, or NPIV) - If the host and library server are both connected to SAN storage, Virtual Machine Manager (VMM) can use a SAN transfer to store the virtual machine in the library. In a SAN transfer, the target LUNs are remapped from the source host to the destination library server. No files are moved, which is why a SAN transfer is much faster than moving virtual machine files from one host to another over a local area network (LAN). VMM can use an NPIV SAN transfer if a host bus adapter (HBA) with NPIV support is available.

- NETWORK TRANSFER - If no faster method is available, VMM uses a network transfer to move the virtual machine files from the host server to the library server over the LAN that connects the two servers. You must use the -SharePath parameter to specify the path to the share in the library where you want to store the virtual machine.

Save-SCVirtualMachine automatically uses the fastest available transfer type. If you want to force a network transfer, you can use the UseLAN parameter. If the host server and library server are the same server, the command will not fail if you specify the UseLAN parameter, but the migration to the library will occur faster if you do not use that parameter.

When a virtual machine is stored in the library, it cannot be started. Before you can start the virtual machine, you must use the Move-SCVirtualMachine cmdlet to deploy it from the library to a host.



---

## Parameters

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-LibraryServer<LibraryServer>**

Specifies a VMM library server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-SharePath<String>**

Specifies a path to a valid library share on an existing library server that uses a Universal Naming Convention (UNC) path.

Example format: `-SharePath "\\LibServer01\LibShare"`

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-UseLAN**

Forces a transfer over the local area network (LAN) even if a faster transfer mechanism, such as a storage area network (SAN) transfer, is available.

|               |       |
|---------------|-------|
| Aliases       | none  |
| Required?     | false |
| Position?     | named |
| Default Value | none  |

|                             |       |
|-----------------------------|-------|
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VM<VM>**

Specifies a virtual machine object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **VirtualMachine**

## **Notes**

- Requires a VMM virtual machine object, which can be retrieved by using the Get-SCVirtualMachine cmdlet.

## **Examples**

### **1: Save a virtual machine to the library.**

The first command gets the virtual machine object named VM01 and stores the object in the \$VM variable.

The second command gets the library server object named LibServer01 and stores the object in the \$Library variable.

---

The last command migrates VM01 from its host and stores it to the location \\LibServer01.Contoso.com\Library01\VMs. The command automatically uses the fastest available transfer type.

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM01"
PS C:\> $LibServer = Get-SCLibraryServer -ComputerName "LibServer01"
PS C:\> Save-SCVirtualMachine -LibraryServer $LibServer -VM $VM -SharePath
"\\LibServer01.Contoso.com\Library01\VMs"
```

## 2: Store a virtual machine in the library asynchronously.

The first command gets the virtual machine object named VM02 and stores the object in the \$VM variable.

The second command gets the library server object named LibServer02 and stores the object in the \$Library variable.

The third command migrates VM02 to the location \\LibServer02.Contoso.com\Library02\VMs. The RunAsynchronously parameter returns control to the command shell immediately and the JobVariable parameter tracks job progress and stores a record of its progress in SaveVMJob. For JobVariable, you do not use the dollar sign (\$) when the variable is created.

The last command displays the contents of \$SaveVMJob.

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM02"
PS C:\> $LibServer = Get-SCLibraryServer -ComputerName "LibServer02"
PS C:\> Save-SCVirtualMachine -LibraryServer $LibServer -VM $VM -SharePath
"\\LibServer02.Contoso.com\Library02\VMs" -RunAsynchronously -JobVariable "SaveVMJob"
PS C:\> $SaveVMJob
```

## 3: Store a virtual machine in the library by forcing a network transfer.

The first command gets the virtual machine object named VM03 and stores the object in the \$VM variable.

The second command gets the library server object named LibServer01 and stores the object in the \$LibServer variable.

The last command stores VM03 to the location \\LibServer01.Contoso.com\Library01\VMs. The UseLAN parameter forces a network transfer over the LAN even if a faster transfer mechanism is available.

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM03"
PS C:\> $LibServer = Get-SCLibraryServer -ComputerName "LibServer01"
PS C:\> Save-SCVirtualMachine -LibraryServer $LibServer -VM $VM -SharePath
"\\LibServer01.Contoso.com\Library01\VMs" -UseLAN
```

## Related topics

[Get-SCVirtualMachine](#)

---

[Move-SCVirtualMachine](#)

[Read-SCVirtualMachine](#)

[Stop-SCVirtualMachine](#)

---

# Set-SCApplicationDeployment

---

## Set-SCApplicationDeployment

Modifies an application deployment.

### Syntax

Parameter Set: Default

```
Set-SCApplicationDeployment [-ApplicationDeployment] <ApplicationDeployment> [-ApplicationPackage <ApplicationPackage> ] [-BlockOnChanges <Boolean> ] [-DACInstanceName <String> ] [-IgnoreDataLoss <Boolean> ] [-JobVariable <String> ] [-Name <String> ] [-PROTipID <Guid> ] [-RollbackOnFailure <Boolean> ] [-RunAsynchronously] [-SkipPolicyValidation <Boolean> ] [-SQLAuthenticationType <String> ] [-SQLDeploymentRunAsAccount <VMMCredential> ] [-SQLInstanceName <String> ] [-UninstallMode <String> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Set-SCApplicationDeployment cmdlet modifies an application deployment.

### Parameters

#### **-ApplicationDeployment<ApplicationDeployment>**

Specifies an application deployment object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

---

## **-ApplicationPackage<ApplicationPackage>**

Specifies an application package object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-BlockOnChanges<Boolean>**

Indicates that the SQL DAC update is blocked if the database schema is different than that defined in the previous DAC.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-DACInstanceName<String>**

Specifies the name of a data-tier application (DAC) instance.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-IgnoreDataLoss<Boolean>**

Indicates that data loss which may occur when updating the SQL Server database is ignored.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-Name<String>**

Specifies the name of a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |



---

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RollbackOnFailure<Boolean>**

Rolls back any changes made if the SQL Server database update fails.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-SkipPolicyValidation<Boolean>**

Indicates whether policy validation against the SQL Server database should occur.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-SQLAuthenticationType<String>**

Specifies the SQL Server authentication type. Valid values are: SQLServerAuthentication, WindowsAuthentication.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-SQLDeploymentRunAsAccount<VMMCredential>**

Specifies a Run As account to use to communicate with a SQL Server deployment.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-SQLInstanceName<String>**

Specifies the name of a SQL Server instance.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-UninstallMode<String>**

Specifies the uninstall mode. Valid values are: MakeUnmanaged, DetachDatabase, DropDatabase.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **ApplicationDeployment**

## Examples

### 1: Update application package for a web application deployment.

The first command gets the application profile object named SvcWebAppProfile01 and stores the object in the \$AppProfile variable.

The second command gets the application deployment object named SvcWebDeployment01 for the application profile stored in \$AppProfile, and then stores the object in the \$AppDeployment variable.

The third command gets the application package object named WebApp02.zip from the VMM library and stores the object in the \$AppPackage variable.

The last command updates the application deployment stored in \$AppDeployment by replacing the previous application package with the one stored in \$AppPackage.

```
PS C:\> $AppProfile = Get-SCApplicationProfile -Name "SvcWebAppProfile01"
```

```
PS C:\> $AppDeployment = Get-SCApplicationDeployment -Name "SvcWebDeployment01" -  
ApplicationProfile $AppProfile
```

```
PS C:\> $AppPackage = Get-SCApplicationPackage -Name "WebApp02.zip"
```

```
PS C:\> Set-SCApplicationDeployment -ApplicationDeployment $AppDeployment -  
ApplicationPackage $AppPackage
```

## Related topics

[Add-SCApplicationDeployment](#)

[Get-SCApplicationDeployment](#)

[Remove-SCApplicationDeployment](#)

---

# Set-SCApplicationHostTemplate

---

## Set-SCApplicationHostTemplate

Configures the properties of an application host template that has been added to a service template.

### Syntax

Parameter Set: DefaultParamSet

```
Set-SCApplicationHostTemplate [-ApplicationHostTemplate] <ApplicationHostTemplate> [-AllowUntrustedServerCertificate <Boolean> ] [-ApplicationProfile <ApplicationProfile> ] [-AuthenticationType <String> ] [-ComputerName <String> ] [-DeploymentOrder <Int32> ] [-Description <String> ] [-JobVariable <String> ] [-Name <String> ] [-Owner <String> ] [-Port <Int32> ] [-PROTipID <Guid> ] [-RunAsAccount <VMMSecurityCredential> ] [-RunAsynchronously] [-ServicingOrder <Int32> ] [-Site <String> ] [-Tag <String> ] [ <CommonParameters> ]
```

### Detailed Description

The Set-SCApplicationHostTemplate cmdlet configures the properties of an application host template that has been added to a service template.

### Parameters

#### **-AllowUntrustedServerCertificate<Boolean>**

Specifies that the deployment may proceed when the target deployment server presents an untrusted server certificate. This parameter is used in conjunction with a Web Application host template.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-ApplicationHostTemplate<ApplicationHostTemplate>**

Specifies an application host template object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-ApplicationProfile<ApplicationProfile>**

Specifies an application profile object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-AuthenticationType<String>**

Specifies the authentication mechanism used to facilitate the deployment of web applications associated with the template. This parameter is used in conjunction with a Web Application host template. Valid values are: NTLM and Basic.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-ComputerName<String>**

Specifies the name of a computer that VMM can uniquely identify on your network. Valid formats are: FQDN, IPv4 or IPv6 address, or NetBIOS name.

NOTE: See the examples for a specific cmdlet to determine how that cmdlet specifies the computer name.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-DeploymentOrder<Int32>**

Specifies the order in which a computer tier, application host, or application is deployed.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-Description<String>**

States a description for the specified object.

|               |       |
|---------------|-------|
| Aliases       | none  |
| Required?     | false |
| Position?     | named |
| Default Value | none  |

---

|                             |       |
|-----------------------------|-------|
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Name<String>**

Specifies the name of a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Owner<String>**

Specifies the owner of a VMM object in the form of a valid domain user account.

Example format: -Owner "Contoso\ReneeLo"

Example format: -Owner "ReneeLo@Contoso"

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | false |



---

|                             |       |
|-----------------------------|-------|
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Port<Int32>**

Specifies the network port to use when adding an object or creating a connection. Valid values are: 1 to 4095.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsAccount<VMMCredential>**

Specifies a Run As account that contains credentials with permission to perform this action.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

---

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-ServicingOrder<Int32>**

Specifies the order in which a computer tier or application host is serviced.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Site<String>**

Specifies the web site to which any associated web deploy packages are deployed. This parameter is used in conjunction with a Web Application host template,

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-Tag<String>**

Specifies a word or phrase to associate with an object so that you can search for all objects with the specified set of tags. You can search for a subset of tags, or you can search for the full set of tags.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **ApplicationHostTemplate**

## **Examples**

### **1: Change the description of the application host template.**

The first command gets the service template object named ServiceTemplate01 and stores the object in the \$ServiceTemplate variable.

The second command gets the application host template object for the service template in \$ServiceTemplate and stores the object in the \$AppHostTemplate variable.

---

The last command changes the description property of the application host template in \$AppHostTemplate.

```
PS C:\> $ServiceTemplate = Get-SCServiceTemplate -Name "ServiceTemplate01"
```

```
PS C:\> $AppHostTemplate = Get-SCApplicationHostTemplate -ServiceTemplate $ServiceTemplate
```

```
PS C:\> Set-SCApplicationHostTemplate -ApplicationHostTemplate $AppHostTemplate -Description  
"This is the updated description"
```

## 2: Change the name of an application host template.

The first command gets the service template object named ServiceTemplate01 and stores the object in the \$ServiceTemplate variable.

The second command gets the application host template object for the service template stored in \$ServiceTemplate and stores the object in the \$AppHostTemplate variable.

The last command changes the name property of the application host template stored in \$AppHostTemplate. Because the application host template is cloned into the service template, renaming the application host template does not affect other service templates in the system.

```
PS C:\> $ServiceTemplate = Get-SCServiceTemplate -Name "ServiceTemplate01"
```

```
PS C:\> $AppHostTemplate = Get-SCApplicationHostTemplate -ServiceTemplate $ServiceTemplate
```

```
PS C:\> Set-SCApplicationHostTemplate -ApplicationHostTemplate $AppHostTemplate -Name "This  
is the updated name"
```

## Related topics

[Add-SCApplicationHostTemplate](#)

[Get-SCApplicationHostTemplate](#)

[Get-SCServiceTemplate](#)

[Remove-SCApplicationHostTemplate](#)

---

# Set-SCApplicationPackage

---

## Set-SCApplicationPackage

Modifies the properties of an application package.

### Syntax

Parameter Set: Default

```
Set-SCApplicationPackage [-ApplicationPackage] <ApplicationPackage[]> [-Description <String> ] [-Enabled <Boolean> ] [-EncryptionKey <PSCredential> ] [-FamilyName <String> ] [-JobVariable <String> ] [-Name <String> ] [-Owner <String> ] [-PROTipID <Guid> ] [-Release <String> ] [-RunAsynchronously] [-SharePath <String> ] [-UserRole <UserRole> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: EquivalencySet

```
Set-SCApplicationPackage [-ApplicationPackage] <ApplicationPackage[]> -FamilyName <String> -Release <String> -SetAsEquivalent [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Set-SCApplicationPackage cmdlet modifies the properties of an application package stored in the VMM library.

### Parameters

#### **-ApplicationPackage<ApplicationPackage[]>**

Specifies an application package object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

---

## **-Description<String>**

States a description for the specified object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-Enabled<Boolean>**

Enables an object when set to \$True, or disables an object when set to \$False. For example, if you want to upgrade software on a virtual machine template, you can disable the template object in the VMM library to temporarily prevent users from using that object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-EncryptionKey<PSCredential>**

Specifies credentials to be used as an encryption key when you add a Hyper-V host located in a perimeter network to VMM.

Example format: -SecurityFile "C:\SecurityFile.txt" -EncryptionKey \$Key

|                        |       |
|------------------------|-------|
| Aliases                | none  |
| Required?              | false |
| Position?              | named |
| Default Value          | none  |
| Accept Pipeline Input? | false |

---

|                             |       |
|-----------------------------|-------|
| Accept Wildcard Characters? | false |
|-----------------------------|-------|

### **-FamilyName<String>**

Specifies a family name for a physical resource in the VMM library. This value is used in conjunction with Release, Namespace, and Type to establish equivalency among library resources.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Name<String>**

Specifies the name of a VMM object.

|               |       |
|---------------|-------|
| Aliases       | none  |
| Required?     | false |
| Position?     | named |
| Default Value | none  |

|                             |       |
|-----------------------------|-------|
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Owner<String>**

Specifies the owner of a VMM object in the form of a valid domain user account.

Example format: -Owner "Contoso\ReneeLo"

Example format: -Owner "ReneeLo@Contoso"

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Release<String>**

Specifies a string that describes the release of a library resource. VMM automatically creates a release value for every resource imported into the library. After the resource has been imported, the string can be customized.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|



|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-SetAsEquivalent**

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-SharePath<String>**

Specifies a path to a valid library share on an existing library server that uses a Universal Naming Convention (UNC) path.

Example format: `-SharePath "\\LibServer01\LibShare"`

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-UserRole<UserRole>**

Specifies a user role object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

---

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **ApplicationPackage**

## Examples

### 1: Update the description for an application package.

The first command gets the application package object named WebApp01.zip and stores the object in the \$AppPackage variable.

The second command updates the description for WebApp01.zip.

```
PS C:\> $AppPackage = Get-SCApplicationPackage -Name "WebApp01.zip"
```

```
PS C:\> Set-SCApplicationPackage -ApplicationPackage $AppPackage -Description "Web application to install on a service."
```

## Related topics

[Get-SCApplicationPackage](#)

[Remove-SCApplicationPackage](#)

---

# Set-SCApplicationProfile

---

## Set-SCApplicationProfile

Modifies the properties of an application profile.

### Syntax

Parameter Set: Default

```
Set-SCApplicationProfile [-ApplicationProfile] <ApplicationProfile> [-CompatibilityType <String> ] [-Description <String> ] [-EnforceCompatibilityType] [-JobVariable <String> ] [-Name <String> ] [-Owner <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-Tag <String> ] [-UserRole <UserRole> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Set-SCApplicationProfile cmdlet modifies the properties of an application profile.

### Parameters

#### **-ApplicationProfile<ApplicationProfile>**

Specifies an application profile object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

#### **-CompatibilityType<String>**

Specifies the deployment types with which an application profile is compatible. Valid values are: General, SQLApplicationHost, WebApplicationHost.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Description<String>**

States a description for the specified object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-EnforceCompatibilityType**

Indicates that artifacts from an application profile which is not compatible with the value provided for the CompatibilityType parameter are removed.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

---

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Name<String>**

Specifies the name of a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Owner<String>**

Specifies the owner of a VMM object in the form of a valid domain user account.

Example format: -Owner "Contoso\ReneeLo"

Example format: -Owner "ReneeLo@Contoso"

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-Tag<String>**

Specifies a word or phrase to associate with an object so that you can search for all objects with the specified set of tags. You can search for a subset of tags, or you can search for the full set of tags.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-UserRole<UserRole>**

Specifies a user role object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## **Examples**

### **1: Change the name of an application profile.**

The first command gets the application profile object named SvcWebAppProfile01 and stores the object in the \$AppProfile variable.

The second command changes the name of the application profile object stored in \$AppProfile to StockWebApp.

```
PS C:\> $AppProfile = Get-SCApplicationProfile -Name "SvcWebAppProfile01"
```

```
PS C:\> Set-SCApplicationProfile -ApplicationProfile $AppProfile -Name "StockWebApp"
```



---

## 2: Change the description of an application profile.

The first command gets the application profile object named StockWebApp and stores the object in the \$AppProfile variable.

The second command changes the description of the application profile object stored in \$AppProfile to "Application profile to use when deploying the stock application web servers."

```
PS C:\> $AppProfile = Get-SCApplicationProfile -Name "StockWebApp"
```

```
PS C:\> Set-SCApplicationProfile -ApplicationProfile $AppProfile -Description "Application profile to use when deploying the stock application web servers."
```

### Related topics

[Get-SCApplicationProfile](#)

[New-SCApplicationProfile](#)

[Remove-SCApplicationProfile](#)

---

# Set-SCApplicationSetting

---

## Set-SCApplicationSetting

Sets the value of an application setting.

### Syntax

Parameter Set: Value

```
Set-SCApplicationSetting [-ApplicationSetting] <ApplicationSetting> [-JobGroup <Guid> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-Value <String> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: SecureValue

```
Set-SCApplicationSetting [-ApplicationSetting] <ApplicationSetting> [-JobGroup <Guid> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-SecureValue <SecureString> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Set-SCApplicationSetting cmdlet sets the value of an application setting.

### Parameters

#### **-ApplicationSetting<ApplicationSetting>**

Specifies an application setting object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

---

## **-JobGroup<Guid>**

Specifies an identifier for a series of commands that will run as a set just before the final command that includes the same job group identifier runs.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-SecureValue<SecureString>**

Specifies the value for a secure string.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-Value<String>**

Specifies a string used to attribute an object or property.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **ApplicationSetting**

## **Examples**

### **1: Set the value for an application setting.**

The first command gets the application profile object named SvcWebAppProfile01 and stores the object in the \$AppProfile variable.

The second command gets the application deployment object named SvcWebDeployment01 for the application profile stored in \$AppProfile and stores the object in the \$AppDeployment variable.

The third command gets the setting object named Order\_Service for the application package in the application deployment stored in \$AppDeployment, and then stores the setting object in the \$AppSetting variable.

The last command sets the value for the application setting stored in \$AppSetting. In this case, the value for the Order\_Service setting was updated in the SvcWebDeployment01 application deployment.

```
PS C:\> $AppProfile = Get-SCApplicationProfile -Name "SvcWebAppProfile01"
```

```
PS C:\> $AppDeployment = Get-SCApplicationDeployment -ApplicationProfile $AppProfile -Name "SvcWebDeployment01"
```

```
PS C:\> $AppSetting = Get-SCApplicationSetting -ApplicationDeployment $AppDeployment -Name "Order_Service"
```

```
PS C:\> Set-SCApplicationSetting -ApplicationSetting $AppSetting -Value "http://@servicesComputerName@/OrderService.xamlx"
```

---

## Related topics

[Get-SCApplicationSetting](#)

---

# Set-SCBaseline

---

## Set-SCBaseline

Modifies a baseline by adding or removing updates or assignment scopes.

### Syntax

Parameter Set: Default

```
Set-SCBaseline [-Baseline] <Baseline> [-AddAssignmentScope <IBaselineAssignmentScope> ] [-AddUpdates <List<SoftwareUpdate>> ] [-Description <String> ] [-JobGroup <Guid> ] [-JobVariable <String> ] [-Name <String> ] [-PROTipID <Guid> ] [-RemoveAssignmentScope <IBaselineAssignmentScope> ] [-RemoveUpdates <List<SoftwareUpdate>> ] [-RunAsynchronously] [-StartNow] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Set-SCBaseline cmdlet modifies a baseline by adding or removing updates or assignment scopes. A baseline is a list of updates which, together with scope assignments, can grade the compliance of required updates for Virtual Machine Manager (VMM) fabric servers.

### Parameters

#### **-AddAssignmentScope<IBaselineAssignmentScope>**

Adds one or more virtual machine hosts or clusters to a baseline assignment scope.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-AddUpdates<List<SoftwareUpdate>>**

Adds one or more software updates to a baseline.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-Baseline<Baseline>**

Specifies a VMM baseline object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-Description<String>**

States a description for the specified object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |



---

## **-JobGroup<Guid>**

Specifies an identifier for a series of commands that will run as a set just before the final command that includes the same job group identifier runs.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-Name<String>**

Specifies the name of a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RemoveAssignmentScope<IBaselineAssignmentScope>**

Removes one or more VMManagedComputer, VMHostCluster, or VMHostGroup objects from a baseline assignment scope.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RemoveUpdates<List<SoftwareUpdate>>**

Removes one or more software updates from a baseline.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-StartNow**

Identifies the last command of a jobgroup and starts running the commands within the jobgroup. This parameter must be used with the JobGroup parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **Baseline**

## Examples

### 1: Modify an existing baseline by adding updates and an assignment scope.

The first command gets the cluster object named Cluster01 and stores the object in the \$Cluster variable.

The second command gets the baseline object named Security Baseline and stores the object in the \$Baseline variable.

The third command gets the security bulletin update object named MS05-051 and stores the object in the \$Update variable.

The last command adds the update stored in \$Update (MS05-051) to the baseline stored in \$Baseline (Security Baseline) and sets the assignment scope to the object stored in \$Cluster (Cluster01).

```
PS C:\> $Cluster = Get-SCVMHostCluster -Name "Cluster01"
PS C:\> $Baseline = Get-SCBaseline -Name "Security Baseline"
PS C:\> $Update = Get-SCUpdate -SecurityBulletinId "MS05-055"
PS C:\> Set-SCBaseline -Baseline $Baseline -AddUpdates $Update -AddAssignmentScope $Cluster
```

### 2: Remove an update and an assignment scope from an existing baseline.

The first command gets the cluster object named Cluster01 and stores the object in the \$Cluster variable.

The second command gets the baseline object named Security Baseline and stores the object in the \$Baseline variable.

The third command gets the security bulletin update object named MS05-051 and stores the object in the \$Update variable.

The last command removes the update stored in \$Update (MS05-051) from the baseline stored in \$Baseline (Security Baseline) and removes the assignment scope stored in \$Cluster (Cluster01).

```
PS C:\> $Cluster = Get-SCVMHostCluster -Name "Cluster01"
PS C:\> $Baseline = Get-SCBaseline -Name "Security Baseline"
PS C:\> $Update = Get-SCUpdate -SecurityBulletinId "MS05-055"
```

---

```
PS C:\> Set-SCBaseline -Baseline $Baseline -RemoveUpdates $Update -RemoveAssignmentScope  
$Cluster
```

## Related topics

[Get-SCBaseline](#)

[New-SCBaseline](#)

[Remove-SCBaseline](#)

---

# Set-SCCapabilityProfile

---

## Set-SCCapabilityProfile

Modifies the properties of a capability profile.

### Syntax

Parameter Set: FromName

```
Set-SCCapabilityProfile -CapabilityProfile <CapabilityProfile> -Name <String> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

Parameter Set: FromValues

```
Set-SCCapabilityProfile -CapabilityProfile <CapabilityProfile> [-CPUCompatibilityModeValue <Boolean> ] [-CPUCompatibilityModeValueCanChange <Boolean> ] [-CPUCountInitial <Int32> ] [-CPUCountMaximum <Int32> ] [-CPUCountMinimum <Int32> ] [-Description <String> ] [-DifferencingVirtualHardDiskValue <Boolean> ] [-DifferencingVirtualHardDiskValueCanChange <Boolean> ] [-DynamicMemoryValue <Boolean> ] [-DynamicMemoryValueCanChange <Boolean> ] [-DynamicVirtualHardDiskValue <Boolean> ] [-DynamicVirtualHardDiskValueCanChange <Boolean> ] [-ExistDiskStorageClassificationValue <Guid> ] [-FixedVirtualHardDiskValue <Boolean> ] [-FixedVirtualHardDiskValueCanChange <Boolean> ] [-JobVariable <String> ] [-LogicalNetworkValue <Guid> ] [-MaximumMemoryMBInitial <Int32> ] [-MaximumMemoryMBMaximum <Int32> ] [-MaximumMemoryMBMinimum <Int32> ] [-MemoryMBInitial <Int32> ] [-MemoryMBMaximum <Int32> ] [-MemoryMBMinimum <Int32> ] [-NetworkOptimizationValue <Boolean> ] [-NetworkOptimizationValueCanChange <Boolean> ] [-NewDiskStorageClassificationValue <Guid> ] [-OSCompatibilityModeValue <Boolean> ] [-OSCompatibilityModeValueCanChange <Boolean> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-SharedDVDImageFileValue <Boolean> ] [-SharedDVDImageFileValueCanChange <Boolean> ] [-StartupMemoryMBInitial <Int32> ] [-StartupMemoryMBMaximum <Int32> ] [-StartupMemoryMBMinimum <Int32> ] [-TargetMemoryBufferPercentInitial <Int32> ] [-TargetMemoryBufferPercentMaximum <Int32> ] [-TargetMemoryBufferPercentMinimum <Int32> ] [-VirtualDVDDriveCountInitial <Int32> ] [-VirtualDVDDriveCountMaximum <Int32> ] [-VirtualDVDDriveCountMinimum <Int32> ] [-VirtualHardDiskCountInitial <Int32> ] [-VirtualHardDiskCountMaximum <Int32> ] [-VirtualHardDiskCountMinimum <Int32> ] [-VirtualHardDiskSizeMBInitial <Int32> ] [-VirtualHardDiskSizeMBMaximum <Int32> ] [-VirtualHardDiskSizeMBMinimum <Int32> ] [-VirtualNetworkAdapterCountInitial <Int32> ] [-VirtualNetworkAdapterCountMaximum <Int32> ] [-VirtualNetworkAdapterCountMinimum <Int32> ] [-VMHighlyAvailableValue <Boolean> ] [-VMHighlyAvailableValueCanChange <Boolean> ] [ <CommonParameters>]
```

### Detailed Description

The Set-SCCapabilityProfile cmdlet modifies the properties of a capability profile object.

For information about creating a capability profile, type: "Get-Help New-SCCapabilityProfile -detailed".

---

## Parameters

### **-CapabilityProfile<CapabilityProfile>**

Specifies a capability profile object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-CPUCompatibilityModeValue<Boolean>**

Indicates whether processor compatibility mode is enabled. When set to \$True, VMM limits the processor features that a virtual machine can use in order to improve compatibility with a different processor version.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-CPUCompatibilityModeValueCanChange<Boolean>**

Indicates whether the value for CPU compatibility mode can be updated.

|               |       |
|---------------|-------|
| Aliases       | none  |
| Required?     | false |
| Position?     | named |
| Default Value | none  |

---

|                             |       |
|-----------------------------|-------|
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-CPUCountInitial<Int32>**

Specifies the initial number of processors that a virtual machine will have when deployed in a private cloud.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-CPUCountMaximum<Int32>**

Specifies the maximum number of processors that a virtual machine deployed in a private cloud can have.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-CPUCountMinimum<Int32>**

Specifies the minimum number of processors that a virtual machine deployed in a private cloud can have.

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | false |



---

|                             |       |
|-----------------------------|-------|
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Description<String>**

States a description for the specified object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-DifferencingVirtualHardDiskValue<Boolean>**

Indicates whether differencing disks are allowed.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-DifferencingVirtualHardDiskValueCanChange<Boolean>**

Indicates whether the value for differencing disks can be updated.

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | false |

---

|                             |       |
|-----------------------------|-------|
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-DynamicMemoryValue<Boolean>**

Indicates whether dynamic memory is enabled.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-DynamicMemoryValueCanChange<Boolean>**

Indicates whether the value for dynamic memory can be updated.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-DynamicVirtualHardDiskValue<Boolean>**

Indicates whether dynamic virtual hard disks are allowed.

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | false |

---

|                             |       |
|-----------------------------|-------|
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-DynamicVirtualHardDiskValueCanChange<Boolean>**

Indicates whether the value for dynamic virtual hard disks can be updated.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-ExistDiskStorageClassificationValue<Guid>**

Specifies a GUID.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-FixedVirtualHardDiskValue<Boolean>**

Indicates whether fixed virtual hard disks are allowed.

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | false |

---

|                             |       |
|-----------------------------|-------|
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-FixedVirtualHardDiskValueCanChange<Boolean>**

Indicates whether the value for fixed virtual hard disks can be updated.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-LogicalNetworkValue<Guid>**

Specifies a GUID.

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | false |

|                             |       |
|-----------------------------|-------|
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-MaximumMemoryMBInitial<Int32>**

Specifies the initial maximum amount of memory, in megabytes (MB), allocated to a virtual machine when deployed in a private cloud.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-MaximumMemoryMBMaximum<Int32>**

Specifies the highest amount of maximum memory, in megabytes (MB), that can be allocated to a virtual machine.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-MaximumMemoryMBMinimum<Int32>**

Specifies the lowest amount of maximum memory, in megabytes (MB), that can be allocated to a virtual machine.

---

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-MemoryMBInitial<Int32>**

Specifies the initial amount of memory, in megabytes (MB), allocated to a virtual machine when deployed in a private cloud.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-MemoryMBMaximum<Int32>**

Specifies the maximum amount of memory, in megabytes (MB), that can be allocated to a virtual machine.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-MemoryMBMinimum<Int32>**

Specifies the minimum amount of memory, in megabytes (MB), that can be allocated to a virtual machine.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-Name<String>**

Specifies the name of a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-NetworkOptimizationValue<Boolean>**

Indicates whether network optimization is enabled.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-NetworkOptimizationValueCanChange<Boolean>**

Indicates whether the value for network optimization can be updated.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-NewDiskStorageClassificationValue<Guid>**

Specifies a GUID.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-OSCompatibilityModeValue<Boolean>**

Indicates whether operating system compatibility mode is enabled.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |



---

## **-OSCompatibilityModeValueCanChange<Boolean>**

Indicates whether the value for operating system compatibility can be updated.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-SharedDVDImageFileValue<Boolean>**

Indicates whether shared DVD image mode is enabled.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-SharedDVDImageFileValueCanChange<Boolean>**

Indicates whether the value for shared DVD image mode can be updated.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-StartupMemoryMBInitial<Int32>**

Specifies the initial amount of memory, in megabytes (MB), that is allocated to a virtual machine upon startup.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

### **-StartupMemoryMBMaximum<Int32>**

Specifies the maximum amount of memory, in megabytes (MB), that is allocated to a virtual machine upon startup.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-StartupMemoryMBMinimum<Int32>**

Specifies the minimum amount of memory, in megabytes (MB), that is allocated to a virtual machine upon startup.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-TargetMemoryBufferPercentInitial<Int32>**

Specifies the initial percentage of memory above a virtual machine's current memory allocation that the host should try to reserve as a buffer.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

### **-TargetMemoryBufferPercentMaximum<Int32>**

Specifies the maximum percentage of memory above a virtual machine's current memory allocation that the host should try to reserve as a buffer.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-TargetMemoryBufferPercentMinimum<Int32>**

Specifies the minimum percentage of memory above a virtual machine's current memory allocation that the host should try to reserve as a buffer.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VirtualDVDDriveCountInitial<Int32>**

Specifies the initial number of DVD drives attached to a virtual machine when deployed in a private cloud.

|               |       |
|---------------|-------|
| Aliases       | none  |
| Required?     | false |
| Position?     | named |
| Default Value | none  |

---

|                             |       |
|-----------------------------|-------|
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VirtualDVDDriveCountMaximum<Int32>**

Specifies the maximum number of DVD drives that can be attached to a virtual machine deployed in a private cloud.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VirtualDVDDriveCountMinimum<Int32>**

Specifies the minimum number of DVD drives that can be attached to a virtual machine deployed in a private cloud.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VirtualHardDiskCountInitial<Int32>**

Specifies the initial number of virtual hard disks attached to a virtual machine when deployed in a private cloud.

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | false |

---

|                             |       |
|-----------------------------|-------|
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VirtualHardDiskCountMaximum<Int32>**

Specifies the maximum number of virtual hard disks that can be attached to a virtual machine deployed in a private cloud.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VirtualHardDiskCountMinimum<Int32>**

Specifies the minimum number of virtual hard disks that can be attached to a virtual machine deployed in a private cloud.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VirtualHardDiskSizeMBInitial<Int32>**

Specifies the initial hard disk size, in megabytes (MB), for a virtual machine when deployed in a private cloud.

---

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VirtualHardDiskSizeMBMaximum<Int32>**

Specifies the maximum virtual hard disk size, in megabytes (MB), allowed for a virtual machine deployed in a private cloud.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VirtualHardDiskSizeMBMinimum<Int32>**

Specifies the minimum virtual hard disk size, in megabytes (MB), allowed for a virtual machine deployed in a private cloud.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

### **-VirtualNetworkAdapterCountInitial<Int32>**

Specifies the initial number of virtual network adapters attached to a virtual machine when deployed in a private cloud.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VirtualNetworkAdapterCountMaximum<Int32>**

Specifies the maximum number of virtual network adapters that can be attached to a virtual machine deployed in a private cloud.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VirtualNetworkAdapterCountMinimum<Int32>**

Specifies the minimum number of virtual network adapters that can be attached to a virtual machine deployed in a private cloud.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |



---

## **-VMHighlyAvailableValue<Boolean>**

Indicates whether a deployed virtual machine will be highly available.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VMHighlyAvailableValueCanChange<Boolean>**

Indicates whether the value indicating the high availability status of a virtual machine can be updated.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Notes**

- Requires a VMM capability profile object, which can be retrieved by using the Get-SCCapabilityProfile cmdlet.

---

## Examples

### 1: Modify the default virtual hard disk settings of a capability profile

The first command gets the capability profile object named CapabilityProf01 and stores the object in the \$CapabilityProfile variable.

The second command sets the virtual hard disk minimum to 1, the virtual hard disk maximum to 8, and the maximum virtual hard disk size to 25600 MB (250 GB) for the capability profile stored in \$CapabilityProfile.

```
PS C:\> $CapabilityProfile = Get-SCCapabilityProfile -Name "CapabilityProf01"
```

```
PS C:\> Set-SCCapabilityProfile -CapabilityProfile $CapabilityProfile -  
VirtualHardDiskCountMinimum 1 -VirtualHardDiskCountMaximum 8 -VirtualHardDiskSizeMBMaximum  
256000
```

### Related topics

[Get-SCCapabilityProfile](#)

[New-SCCapabilityProfile](#)

[Remove-SCCapabilityProfile](#)

[Test-SCCapabilityProfile](#)

---

# Set-SCCloud

---

## Set-SCCloud

Changes the properties of a private cloud in VMM.

### Syntax

Parameter Set: Default

```
Set-SCCloud [[-Cloud] <Cloud> ] [-AddCapabilityProfile <List<CapabilityProfile>> ] [-AddCloudResource <List<ClientObject>> ] [-AddReadOnlyLibraryShare <List<LibraryShare>> ] [-AddVMHostGroup <List<HostGroup>> ] [-Description <String> ] [-JobGroup <Guid> ] [-JobVariable <String> ] [-Name <String> ] [-PROTipID <Guid> ] [-ReadWriteLibraryPath <String> ] [-RemoveCapabilityProfile <List<CapabilityProfile>> ] [-RemoveCloudResource <List<ClientObject>> ] [-RemoveReadOnlyLibraryShare <List<LibraryShare>> ] [-RemoveVMHostGroup <List<HostGroup>> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Set-SCCloud cmdlet changes the properties of a private cloud in Virtual Machine Manager (VMM). For more information about private clouds, type: "Get-Help New-SCCloud -detailed".

### Parameters

#### **-AddCapabilityProfile<List<CapabilityProfile>>**

Adds one or more capability profile objects to a private cloud.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-AddCloudResource<List<ClientObject>>**

Adds one or more resources to a private cloud. Resources that you can add to a private cloud include: Load Balancer, Logical Network, Static IP Address Pool, Storage Pool, and VIP Template.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-AddReadOnlyLibraryShare<List<LibraryShare>>**

Adds one or more specified read-only library shares to a private cloud.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-AddVMHostGroup<List<HostGroup>>**

Adds one or more host groups to an existing host group array or private cloud.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-Cloud<Cloud>**

Specifies a private cloud object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-Description<String>**

States a description for the specified object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-JobGroup<Guid>**

Specifies an identifier for a series of commands that will run as a set just before the final command that includes the same job group identifier runs.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-Name<String>**

Specifies the name of a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-ReadWriteLibraryPath<String>**

Specifies a writable library path to which virtual machines can be stored.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RemoveCapabilityProfile<List<CapabilityProfile>>**

Removes one or more specified capability profile objects.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RemoveCloudResource<List<ClientObject>>**

Removes a specified resource from a private cloud.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-RemoveReadOnlyLibraryShare<List<LibraryShare>>**

Removes a specified read-only library share.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RemoveVMHostGroup<List<HostGroup>>**

Removes one or more host groups from a host group array or private cloud.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |



---

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Examples**

### **1: Update the properties of a private cloud using a job group.**

The first command creates a new GUID and stores it in the \$Guid variable. Subsequent commands that include this GUID are collected into a single job group.

The second command gets the private cloud object named Cloud01 and stores the object in the \$Cloud variable.

The third command gets the Hyper-V capability profile object and stores the object in the \$CapabilityProfile variable.

The fourth command sets the following limits on the cloud capacity dimensions: 500 GB storage, 50 virtual machines, and 100 CPUs. Using the JobGroup parameter specifies that this command will not run until just before the final command that includes the JobGroup with the same GUID.

The last command adds the capability profile stored in \$CapabilityProfile to Cloud01, updates the description for Cloud01, and sets the cloud capacity properties on Cloud01 using the settings specified in the fourth command. This command uses the JobGroup parameter to run Set-SCCloudCapacity just before Set-SCCloud runs so that the settings will be associated with the specified private cloud.

```
PS C:\> $Guid = [System.Guid]::NewGuid()
```

```
PS C:\> $Cloud = Get-SCCloud -Name "Cloud01"
```

```
PS C:\> $CapabilityProfile = Get-SCCapabilityProfile -ProfileName "HyperV"
```

```
PS C:\> Set-SCCloudCapacity -JobGroup $Guid -StorageLimited $True -StorageGB 500 -  
VirtualMachinesLimited $True -VirtualMachines 50 -VirtualCPUCountLimited $True -  
VirtualCPUCount 100
```

```
PS C:\> Set-SCCloud -Cloud $Cloud -AddCapabilityProfile $CapabilityProfile -JobGroup $Guid -  
Description "Updated description for Cloud01"
```

---

## Related topics

[Get-SCCapabilityProfile](#)

[Get-SCCloud](#)

[New-SCCloud](#)

[Remove-SCCloud](#)

[Set-SCCloudCapacity](#)

---

# Set-SCCloudCapacity

---

## Set-SCCloudCapacity

Modifies the cloud capacity settings for a private cloud.

### Syntax

Parameter Set: FromValues

```
Set-SCCloudCapacity -CloudCapacity <CloudCapacity> [-CPUCount <UInt32> ] [-CustomQuotaCount <UInt32> ] [-JobVariable <String> ] [-MemoryMB <UInt32> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-StorageGB <UInt32> ] [-UseCPUCountMaximum <Boolean> ] [-UseCustomQuotaCountMaximum <Boolean> ] [-UseMemoryMBMaximum <Boolean> ] [-UseStorageGBMaximum <Boolean> ] [-UseVMCountMaximum <Boolean> ] [-VMCount <UInt32> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: JobGroupParamSet

```
Set-SCCloudCapacity -JobGroup <Guid> [-CPUCount <UInt32> ] [-CustomQuotaCount <UInt32> ] [-JobVariable <String> ] [-MemoryMB <UInt32> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-StorageGB <UInt32> ] [-UseCPUCountMaximum <Boolean> ] [-UseCustomQuotaCountMaximum <Boolean> ] [-UseMemoryMBMaximum <Boolean> ] [-UseStorageGBMaximum <Boolean> ] [-UseVMCountMaximum <Boolean> ] [-VMCount <UInt32> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Set-SCCloudCapacity cmdlet modifies the cloud capacity settings for a private cloud in Virtual Machine Manager (VMM). You can update the following cloud capacity dimensions:

- virtual machines
- virtual CPUs
- custom quota points
- storage (GB)
- memory (MB)

Alternatively, you can set any or all of the dimensions to use the maximum capacity.

### Parameters

#### **-CloudCapacity<CloudCapacity>**

Specifies a cloud capacity object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-CPUCount<UInt32>**

Specifies the number of virtual CPUs for a user role quota or cloud capacity.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-CustomQuotaCount<UInt32>**

Specifies the number of custom quota points for a user role quota or cloud capacity.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobGroup<Guid>**

Specifies an identifier for a series of commands that will run as a set just before the final command that includes the same job group identifier runs.

---

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-MemoryMB<UInt32>**

Specifies the amount of memory in megabytes (MB) for a user role quota or cloud capacity.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-StorageGB<UInt32>**

Specifies the amount of storage in gigabytes (GB) for a user role quota or cloud capacity. This storage amount does not include library storage.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

### **-UseCPUCountMaximum<Boolean>**

Indicates that the maximum number of virtual CPUs is allowed for a user role or cloud capacity. When this parameter is used, no quota is enforced for the virtual CPU dimension.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-UseCustomQuotaCountMaximum<Boolean>**

Indicates that the maximum number of custom quota points is allowed for a user role or cloud capacity. When this parameter is used, no quota is enforced for the custom quota dimension.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-UseMemoryMBMaximum<Boolean>**

Indicates that the maximum amount of memory, in megabytes (MB), is allowed for a user role or cloud capacity. When this parameter is used, no quota is enforced for the memory dimension.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

### **-UseStorageGBMaximum<Boolean>**

Indicates that the maximum amount of storage, in gigabytes (GB), is allowed for a user role or cloud capacity. When this parameter is used, no quota is enforced for the storage dimension.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-UseVMCountMaximum<Boolean>**

Indicates that the maximum number of virtual machines is allowed for a user role or cloud capacity. When this parameter is used, no quota is enforced for the virtual machine dimension.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMCount<UInt32>**

Specifies the number of virtual machines for a user role quota or cloud capacity.

|                        |       |
|------------------------|-------|
| Aliases                | none  |
| Required?              | false |
| Position?              | named |
| Default Value          | none  |
| Accept Pipeline Input? | false |



|                             |       |
|-----------------------------|-------|
| Accept Wildcard Characters? | false |
|-----------------------------|-------|

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Examples**

### **1: Change the cloud capacity properties of a specified cloud.**

The first command gets the private cloud object named Cloud01 and stores the object in the \$Cloud variable.

The second command gets the cloud capacity for the private cloud stored in \$Cloud and stores it in the \$CloudCapacity variable.

The last command indicates that there should be a limit placed on the virtual CPU count for the cloud capacity stored in \$CloudCapacity, and changes the virtual CPU count capacity to 20.

```
PS C:\> $Cloud = Get-SCCloud -Name "Cloud01"
```

```
PS C:\> $CloudCapacity = Get-SCCloudCapacity -Cloud $Cloud
```

```
PS C:\> Set-SCCloudCapacity -CloudCapacity $CloudCapacity -VirtualCPUCountLimited $True -VirtualCPUCount 20
```

### **2: Change the cloud capacity properties of a specific private cloud using a job group.**

The first command creates a new GUID and stores it in the \$Guid variable. Subsequent commands that include this GUID are collected into a single job group.

---

The second command gets the private cloud object named Cloud02 and stores the object in the \$Cloud variable.

The third command gets the cloud capacity object for the private cloud stored in \$cloud and stores the object in the \$CloudCapacity variable.

The fourth command sets a limit of 50 virtual machines, 100 virtual CPUs and 500 GB of storage on the cloud capacity. Using the JobGroup parameter specifies that this command will not run until just before the final command that includes the JobGroup with the same GUID.

The last command sets the capacity properties on the private cloud stored in \$Cloud using the settings that were specified in the fourth command. This command uses the JobGroup parameter to run Set-SCCloudCapacity just before Set-SCCloud runs so that the settings will be associated with the specified private cloud.

```
PS C:\> $Guid = [System.Guid]::NewGuid()
```

```
PS C:\> $Cloud = Get-SCCloud -Name "Cloud02"
```

```
PS C:\> $CloudCapacity = Get-SCCloudCapacity -Cloud $Cloud
```

```
PS C:\> Set-SCCloudCapacity -JobGroup $Guid -VirtualMachinesLimited $True -VirtualMachines  
50 -VirtualCPUCountLimited $True -VirtualCPUCount 100 -StorageLimited $True -StorageGB 500
```

```
PS C:\> Set-SCCloud -JobGroup $Guid -Cloud $Cloud
```

## Related topics

[Get-SCCloud](#)

[Get-SCCloudCapacity](#)

---

# Set-SCComplianceStatus

---

## Set-SCComplianceStatus

Modifies a compliance status object.

### Syntax

Parameter Set: Default

```
Set-SCComplianceStatus [-ComplianceStatus] <ComplianceStatus> -Baseline <Baseline> -Update <SoftwareUpdate> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: AddExemption

```
Set-SCComplianceStatus [-ComplianceStatus] <ComplianceStatus> -AddExemption -Baseline <Baseline> -Update <SoftwareUpdate> [-ExemptionNote <String> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: RemoveExemption

```
Set-SCComplianceStatus [-ComplianceStatus] <ComplianceStatus> -Baseline <Baseline> -RemoveExemption -Update <SoftwareUpdate> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Set-SCComplianceStatus cmdlet modifies a compliance status object.

### Parameters

#### -AddExemption

Adds an exemption to an update that is part of a baseline.

|                        |       |
|------------------------|-------|
| Aliases                | none  |
| Required?              | true  |
| Position?              | named |
| Default Value          | none  |
| Accept Pipeline Input? | false |

|                             |       |
|-----------------------------|-------|
| Accept Wildcard Characters? | false |
|-----------------------------|-------|

### **-Baseline<Baseline>**

Specifies a VMM baseline object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-ComplianceStatus<ComplianceStatus>**

Specifies a compliance status object. The compliance status of an object indicates the object's compliance to the baselines to which the object is assigned.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-ExemptionNote<String>**

States a business reason for the exempted update.

|               |       |
|---------------|-------|
| Aliases       | none  |
| Required?     | false |
| Position?     | named |
| Default Value | none  |

---

|                             |       |
|-----------------------------|-------|
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RemoveExemption**

Removes an exemption from an update that is part of a baseline.

|               |       |
|---------------|-------|
| Aliases       | none  |
| Required?     | true  |
| Position?     | named |
| Default Value | none  |

|                             |       |
|-----------------------------|-------|
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Update<SoftwareUpdate>**

Specifies a software update object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|               |       |
|---------------|-------|
| Aliases       | none  |
| Required?     | false |
| Position?     | named |
| Default Value | none  |

---

|                             |                |
|-----------------------------|----------------|
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **ComplianceStatus**

## Examples

### 1: Add an exemption to a compliance status.

The first command gets the host object named VMHost01 and stores the object in the \$VMHost variable.

The second command gets the compliance status for VMHost01 and stores the status object in the \$Compliance variable.

The third command gets the baseline named Security Baseline and stores the object in the \$Baseline variable.

The fourth command gets the security bulletin update MS05-055 and stores the update object in the \$Update variable.

The last command adds an exemption to the update MS05-055 that is part of the Security Baseline baseline, and an exemption note with a business reason for the exemption.

```
PS C:\> $VMHost = Get-SCVMHost -ComputerName "VMHost01"
```

```
PS C:\> $Compliance = Get-SCComplianceStatus -VMMManagedComputer $VMHost.ManagedComputer
```

```
PS C:\> $Baseline = Get-SCBaseline -Name "Security Baseline"
```

```
PS C:\> $Update = Get-SCUpdate -SecurityBulletinID "MS05-055"
```

```
PS C:\> Set-SCComplianceStatus -ComplianceStatus $Compliance -Baseline $Baseline -Update $Update -AddExemption -ExemptionNote "This exemption has been signed off by the IT Manager."
```

## Related topics

[Get-SCComplianceStatus](#)

---

# Set-SCComputerTier

---

## Set-SCComputerTier

Modifies the properties of a VMM computer tier object.

### Syntax

Parameter Set: Default

```
Set-SCComputerTier -ComputerTier <ComputerTier> [-AvailabilitySetName <String> ] [-Description <String> ] [-InstanceMaximumCount <Int32> ] [-InstanceMinimumCount <Int32> ] [-JobVariable <String> ] [-Name <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-ServicingType {UseStandardServicing | UseImageBasedServicing} ] [ <CommonParameters>]
```

### Detailed Description

The Set-SCComputerTier cmdlet modifies the properties of a Virtual Machine Manager (VMM) computer tier object.

### Parameters

#### **-AvailabilitySetName<String>**

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### **-ComputerTier<ComputerTier>**

Specifies a computer tier object.



---

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-Description<String>**

States a description for the specified object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-InstanceMaximumCount<Int32>**

Specifies the maximum number of virtual machines to which a service instance can scale out.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-InstanceMinimumCount<Int32>**

Specifies the minimum number of virtual machines to which a service instance can scale in.

---

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Name<String>**

Specifies the name of a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-ServicingType<ServicingTypeValues>**

Specifies the type of servicing for a service. Valid values are: UseStandardServicing, UseImageBasedServicing.

The acceptable values for this parameter are:

|                        |  |
|------------------------|--|
| UseStandardServicing   |  |
| UseImageBasedServicing |  |

|                        |       |
|------------------------|-------|
| Aliases                | none  |
| Required?              | false |
| Position?              | named |
| Default Value          | none  |
| Accept Pipeline Input? | false |

---

|                             |
|-----------------------------|
| Accept Wildcard Characters? |
|-----------------------------|

|       |
|-------|
| false |
|-------|

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **ComputerTier**

## Examples

### 1: Set the maximum virtual machine count for a computer tier.

The first command gets the service object named Service01 and stores the object in the \$Service variable.

The second command gets the computer tier for the service stored in \$Service and stores the object in the \$Tier variable.

The last command sets the maximum virtual machine count for the computer tier stored in \$Tier to 10.

```
PS C:\> $Service = Get-SCService -Name "Service01"
```

```
PS C:\> $Tier = Get-SCComputerTier -Service $Service
```

```
PS C:\> Set-SCComputerTier -ComputerTier $Tier -InstanceMaximumCount 10
```

## Related topics

[Get-SCComputerTier](#)

---

# Set-SCComputerTierConfiguration

---

## Set-SCComputerTierConfiguration

Configures the computer tier configuration object in an undeployed service configuration.

### Syntax

Parameter Set: Default

```
Set-SCComputerTierConfiguration -ComputerTierConfiguration <BaseComputerTierConfiguration> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-TimeZone <Int32> ] [ <CommonParameters>]
```

### Detailed Description

The Set-SCComputerTierConfiguration cmdlet configures the computer tier configuration object in an undeployed service configuration.

### Parameters

#### **-ComputerTierConfiguration<BaseComputerTierConfiguration>**

Specifies a computer tier configuration object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-TimeZone<Int32>**

Specifies a number (an index) that identifies a geographical region that shares the same standard time. For a list of time zone indexes, see "Microsoft Time Zone Index Values" at:

---

<http://go.microsoft.com/fwlink/?LinkId=120935>. If no time zone is specified, the default time zone used for a virtual machine is the same time zone setting that is on the virtual machine host.

Example format to specify the GMT Standard Time zone: -TimeZone 085

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **ComputerTierConfiguration**

## Examples

### 1: Configure the computer tier within a service configuration.

The first command gets the service configuration object named Service01 and stores the object in the \$ServiceConfig variable.

The second command gets the computer tier configuration for the service configuration stored in \$ServiceConfig and stores the object in the \$TierConfig variable.

The third command sets the timezone property for the computer tier configuration stored in \$TierConfig and stores the computer tier configuration in the \$UpdatedConfig variable.

The last command displays information about the updated computer tier configuration stored in \$UpdatedConfig to the user.

```
PS C:\> $ServiceConfig = Get-SCServiceConfiguration -Name "Service01"
```

```
PS C:\> $TierConfig = Get-SCComputerTierConfiguration -ServiceConfiguration $ServiceConfig
```

```
PS C:\> $UpdatedConfig = Set-SCComputerTierConfiguration -ComputerTierConfiguration  
$TierConfig -TimeZone 085
```

```
PS C:\> $UpdatedConfig
```

---

## Related topics

[Get-SCComputerTierConfiguration](#)



---

# Set-SCComputerTierTemplate

---

## Set-SCComputerTierTemplate

Modifies the properties of a computer tier template.

### Syntax

Parameter Set: Default

```
Set-SCComputerTierTemplate -ComputerTierTemplate <ComputerTierTemplate> [-BlockAutomaticMigration <Boolean> ] [-DefaultInstanceCount <Int32> ] [-DeploymentOrder <Int32> ] [-Description <String> ] [-EnableAvailabilitySet <Boolean> ] [-InstanceMaximumCount <Int32> ] [-InstanceMinimumCount <Int32> ] [-JobVariable <String> ] [-Name <String> ] [-NumberOfUpgradeDomains <Int32> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-ServicingOrder <Int32> ] [-Tag <String> ] [-VMTemplate <Template> ] [ <CommonParameters>]
```

### Detailed Description

The Set-SCComputerTierTemplate cmdlet modifies the properties of a computer tier template.

### Parameters

#### **-BlockAutomaticMigration<Boolean>**

Indicates whether the computer can be automatically migrated. When set to \$True, automatic migration is blocked. When set to \$False, automatic migration is allowed. Default value: \$False.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-ComputerTierTemplate<ComputerTierTemplate>**

Specifies a computer tier template object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-DefaultInstanceCount<Int32>**

Specifies the default instance count for a computer tier that can be scaled out. Default value: 1.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-DeploymentOrder<Int32>**

Specifies the order in which a computer tier, application host, or application is deployed.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

### **-Description<String>**

States a description for the specified object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-EnableAvailabilitySet<Boolean>**

Indicates whether an availability set is generated on a service tier when a service is deployed.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-InstanceMaximumCount<Int32>**

Specifies the maximum number of virtual machines to which a service instance can scale out.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-InstanceMinimumCount<Int32>**

Specifies the minimum number of virtual machines to which a service instance can scale in.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-Name<String>**

Specifies the name of a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-NumberOfUpgradeDomains<Int32>**

Specifies the number of upgrade domains for a computer tier that can be scaled out. Default value: 1.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-ServicingOrder<Int32>**

Specifies the order in which a computer tier or application host is serviced.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-Tag<String>**

Specifies a word or phrase to associate with an object so that you can search for all objects with the specified set of tags. You can search for a subset of tags, or you can search for the full set of tags.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VMTemplate<Template>**

Specifies a VMM template object used to create virtual machines.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **ComputerTierTemplate**

## Examples

### 1: Set the properties of a computer tier template.

The first command gets the service template object named ServiceTemplate01 and stores the object in the \$ServiceTemplate variable.

The second command gets the computer tier template object for the service template stored in \$ServiceTemplate.

The last command sets properties for the computer template tier object stored in \$TierTemplate.

```
PS C:\> $ServiceTemplate = Get-SCServiceTemplate -Name "ServiceTemplate01"
```

```
PS C:\> $TierTemplate = Get-SCComputerTierTemplate -ServiceTemplate $ServiceTemplate
```

```
PS C:\> Set-SCComputerTierTemplate -ComputerTierTemplate $TierTemplate -DefaultInstanceCount  
2 -InstanceMinimumCount 1
```

## Related topics

[Add-SCComputerTierTemplate](#)

[Get-SCComputerTierTemplate](#)

[Get-SCServiceTemplate](#)

[Remove-SCComputerTierTemplate](#)

---

# Set-SCCustomPlacementRule

---

## Set-SCCustomPlacementRule

Modifies a custom placement rule in the placement configuration of a host group.

### Syntax

Parameter Set: MustMatch

```
Set-SCCustomPlacementRule -CustomPlacementRule <CustomPlacementRule> -MustMatch [-  
JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer  
<ServerConnection> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: MustNotMatch

```
Set-SCCustomPlacementRule -CustomPlacementRule <CustomPlacementRule> -MustNotMatch [-  
JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer  
<ServerConnection> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ShouldMatch

```
Set-SCCustomPlacementRule -CustomPlacementRule <CustomPlacementRule> -ShouldMatch [-  
JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer  
<ServerConnection> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ShouldNotMatch

```
Set-SCCustomPlacementRule -CustomPlacementRule <CustomPlacementRule> -ShouldNotMatch [-  
JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer  
<ServerConnection> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The Set-SCCustomPlacementRule modifies an existing custom placement rule in the placement configuration for a host group.

### Parameters

#### **-CustomPlacementRule<CustomPlacementRule>**

Specifies a custom placement rule object.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|



|                             |                |
|-----------------------------|----------------|
| Required?                   | true           |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-MustMatch**

Indicates that the property value of the virtual machine must match the host.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-MustNotMatch**

Indicates that the property value of the virtual machine must not match the host.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

---

|                             |       |
|-----------------------------|-------|
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-ShouldMatch**

Indicates that the property value of the virtual machine should match the host.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

---

|                             |       |
|-----------------------------|-------|
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-ShouldNotMatch**

Indicates that the property value of the virtual machine should not match the host.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Confirm**

Prompts you for confirmation before executing the command.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                             |       |
|-----------------------------|-------|
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## -WhatIf

Describes what would happen if you executed the command without actually executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Examples

### 1: Modify an existing custom placement rule in the placement configuration for a host group.

The first command gets the host group object named HostGroup01 and stores the object in the \$HostGroup variable.

The second command gets the placement configuration object for the host group stored in \$HostGroup and places the object in the \$PlacementConfig variable.

The third command gets the custom placement rule object named Charge Code and stores the object in the \$CPRule variable

The last command modifies the custom placement rule for custom property Charge Code to be a Must Match rule

```
PS C:\> $HostGroup = Get-SCVMHostGroup "HostGroup01"
```

```
PS C:\> $PlacementConfig = Get-SCPlacementConfiguration -VMHostGroup $HostGroup
```

```
PS C:\> $CPRule = Get-SCCustomPlacementRule -PlacementConfiguration $PlacementConfig | where {$_.CustomPropertyName -eq "Charge Code"}
```

---

```
PS C:\> Set-SCCustomPlacementRule -MustMatch -CustomPlacementRule $CPRule
```

## Related topics

[Get-SCCustomPlacementRule](#)

[Remove-SCCustomPlacementRule](#)

---

# Set-SCCustomProperty

---

## Set-SCCustomProperty

Modifies the properties of a custom property.

### Syntax

Parameter Set: \_\_AllParameterSets

```
Set-SCCustomProperty -CustomProperty <CustomProperty> [-Description <String> ] [-JobVariable <String> ] [-Name <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: AddMembers

```
Set-SCCustomProperty -AddMember {VM | VMHost | Template | ServiceTemplate | ServiceInstance | Cloud | VMHostGroup | HostCluster | ComputerTier} -CustomProperty <CustomProperty> [-Description <String> ] [-JobVariable <String> ] [-Name <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: RemoveMembers

```
Set-SCCustomProperty -CustomProperty <CustomProperty> -RemoveMember {VM | VMHost | Template | ServiceTemplate | ServiceInstance | Cloud | VMHostGroup | HostCluster | ComputerTier} [-Description <String> ] [-JobVariable <String> ] [-Name <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Set-SCCustomProperty cmdlet modifies the properties of a custom property. Properties that can be modified include the following:

- Description of the custom property
- Name of the custom property
- Add a member to the custom property
- Remove a member from the custom property

For information about creating a custom property, type: "Get-Help New-SCCustomProperty -detailed".

### Parameters

#### **-AddMember<CustomPropertyObjectType[]>**

Adds one or more members to an object that has the concept of members, such as a group. For example, AddMember adds one or more Active Directory domain users or groups to a user role.

---

Example formats:

-AddMember Domain\User

-AddMember User

-AddMember User@Domain

-AddMember Domain\LabGroupAlias

-AddMember LabGroupAlias (an Active Directory security group, not an email alias)

The acceptable values for this parameter are:

|                 |  |
|-----------------|--|
| VM              |  |
| VMHost          |  |
| Template        |  |
| ServiceTemplate |  |
| ServiceInstance |  |
| Cloud           |  |
| VMHostGroup     |  |
| HostCluster     |  |
| ComputerTier    |  |

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-CustomProperty<CustomProperty>**

Specifies a custom property object.

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | true  |
| Position? | named |

---

|                             |                |
|-----------------------------|----------------|
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-Description<String>**

States a description for the specified object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Name<String>**

Specifies the name of a VMM object.

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | false |
| Position? | named |



|                             |       |
|-----------------------------|-------|
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RemoveMember<CustomPropertyObjectType[]>**

Removes a member from a VMM object that has the concept of membership, such as a group. For example, RemoveMember removes one or more Active Directory domain users or groups from a user role.

Example formats:

-RemoveMember Domain\User

-RemoveMember User

-RemoveMember User@Domain

-RemoveMember Domain\LabGroupAlias

-RemoveMember LabGroupAlias (an Active Directory security group, not an email alias)

The acceptable values for this parameter are:

|                 |  |
|-----------------|--|
| VM              |  |
| VMHost          |  |
| Template        |  |
| ServiceTemplate |  |
| ServiceInstance |  |
| Cloud           |  |

---

|              |  |
|--------------|--|
| VMHostGroup  |  |
| HostCluster  |  |
| ComputerTier |  |

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                        |                |
|------------------------|----------------|
| Aliases                | none           |
| Required?              | false          |
| Position?              | named          |
| Default Value          | none           |
| Accept Pipeline Input? | True (ByValue) |

---

|                             |
|-----------------------------|
| Accept Wildcard Characters? |
|-----------------------------|

|       |
|-------|
| false |
|-------|

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Examples

### 1: Add a member to a custom property.

The first command gets the custom property object named Cost Center and stores the object in the \$CustomProp variable.

The second command adds the VMHost member to the custom property stored in \$CustomProp.

```
PS C:\> $CustomProp = Get-SCCustomProperty -Name "Cost Center"
```

```
PS C:\> Set-SCCustomProperty -CustomProperty $CustomProp -AddMember "VMHost"
```

### 2: Remove a member from a custom property.

The first command gets the custom property object named Cost Center and stores the object in the \$CustomProp variable.

The second command removes the VM member from the custom property object stored in \$CustomProp.

```
PS C:\> $CustomProp = Get-SCCustomProperty -Name "Cost Center"
```

```
PS C:\> Set-SCCustomProperty -CustomProperty $CustomProp -RemoveMember "VM"
```

## Related topics

[Get-SCCustomProperty](#)

[New-SCCustomProperty](#)

[Remove-SCCustomProperty](#)

---

# Set-SCCustomPropertyValue

---

## Set-SCCustomPropertyValue

Updates the value of a custom property.

### Syntax

Parameter Set: InputObject

```
Set-SCCustomPropertyValue -CustomProperty <CustomProperty> -InputObject <ClientObject> -  
Value <String> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [  
<CommonParameters>]
```

Parameter Set: JobGroup

```
Set-SCCustomPropertyValue -CustomProperty <CustomProperty> -JobGroup <Guid> -Value <String>  
[-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

### Detailed Description

The Set-SCCustomPropertyValue cmdlet updates the value of a custom property.

### Parameters

#### **-CustomProperty<CustomProperty>**

Specifies a custom property object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

---

## **-InputObject<ClientObject>**

Specifies the object that is assigned the property whose value you want to retrieve or change.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-JobGroup<Guid>**

Specifies an identifier for a series of commands that will run as a set just before the final command that includes the same job group identifier runs.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-Value<String>**

Specifies a string used to attribute an object or property.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Notes

- Requires a VMM custom property object, which can be retrieved by using the Get-SCCustomProperty cmdlet.

## Examples

### 1: Set the value for a custom property on a virtual machine.

The first command gets the virtual machine object named VM01 and stores the object in the \$VM variable.

The second command gets the custom property object named Cost Center and stores the object in the \$CustomProp variable.

The last command sets the value for the custom property stored in \$CustomProp (Cost Center) for the virtual machine stored in \$VM (VM01) to 123.

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM01"
```

```
PS C:\> $CustomProp = Get-SCCustomProperty -Name "Cost Center"
```

```
PS C:\> Set-SCCustomPropertyValue -InputObject $VM -CustomProperty $CustomProp -Value "123"
```

## Related topics

[Get-SCCustomProperty](#)

[Get-SCCustomPropertyValue](#)

[Remove-SCCustomPropertyValue](#)

---

# Set-SCCustomResource

---

## Set-SCCustomResource

Sets the properties of a custom resource.

### Syntax

Parameter Set: Default

```
Set-SCCustomResource [-CustomResource] <CustomResource[]> [-Description <String> ] [-Enabled <Boolean> ] [-FamilyName <String> ] [-JobVariable <String> ] [-Name <String> ] [-Owner <String> ] [-PROTipID <Guid> ] [-Release <String> ] [-RunAsynchronously] [-SharePath <String> ] [-UserRole <UserRole> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: EquivalencySet

```
Set-SCCustomResource [-CustomResource] <CustomResource[]> -FamilyName <String> -Release <String> -SetAsEquivalent [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Set-SCCustomResource cmdlet sets the properties of a custom resource. A custom resource is a folder-based library object in Virtual Machine Manager (VMM). The resource is declared at the folder level, and the contents of the folder is unknown to VMM.

To add a custom resource to the VMM library, create a folder with a .CR extension, place content in the folder, and then use the VMM console to drag the folder to a VMM library share. VMM discovers and imports the folder into the VMM library as a custom resource.

### Parameters

#### **-CustomResource<CustomResource[]>**

Specifies a custom resource object.

|               |      |
|---------------|------|
| Aliases       | none |
| Required?     | true |
| Position?     | 1    |
| Default Value | none |



|                             |                |
|-----------------------------|----------------|
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-Description<String>**

States a description for the specified object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Enabled<Boolean>**

Enables an object when set to \$True, or disables an object when set to \$False. For example, if you want to upgrade software on a virtual machine template, you can disable the template object in the VMM library to temporarily prevent users from using that object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-FamilyName<String>**

Specifies a family name for a physical resource in the VMM library. This value is used in conjunction with Release, Namespace, and Type to establish equivalency among library resources.

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | false |

---

|                             |       |
|-----------------------------|-------|
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Name<String>**

Specifies the name of a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Owner<String>**

Specifies the owner of a VMM object in the form of a valid domain user account.

Example format: -Owner "Contoso\ReneeLo"

Example format: -Owner "ReneeLo@Contoso"

---

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Release<String>**

Specifies a string that describes the release of a library resource. VMM automatically creates a release value for every resource imported into the library. After the resource has been imported, the string can be customized.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-SetAsEquivalent**

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-SharePath<String>**

Specifies a path to a valid library share on an existing library server that uses a Universal Naming Convention (UNC) path.

Example format: `-SharePath "\\LibServer01\LibShare"`

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-UserRole<UserRole>**

Specifies a user role object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **CustomResource**

---

## Examples

### 1: Change the description and release of a custom resource.

The first command gets the custom resource object named Folder.CR on LibraryServer01 from the VMM library on VMMServer01 and then stores the object in the \$CR variable.

The second command changes the values for the Release and Description properties of the custom resource object stored in \$CR,.

```
PS C:\> $CR = Get-SCCustomResource -VMMServer "VMMServer01.Contoso.com" | where { $_.Name -eq "Folder.CR" -and $_.LibraryServer.Name -eq "LibraryServer01.Contoso.com" }
```

```
PS C:\> Set-SCCustomResource -CustomResource $CR -Release "v1.1" -Description "My LOB Application version 1.1 Install Package"
```

## Related topics

[Get-SCCustomResource](#)

[Remove-SCCustomResource](#)

---

# Set-SCDriverPackage

---

## Set-SCDriverPackage

Updates the properties of an existing driver package object.

### Syntax

Parameter Set: Default

```
Set-SCDriverPackage [-DriverPackage] <DriverPackage> -Tag <List<String>> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Set-SCDriverPackage cmdlet updates the properties of an existing driver package object.

### Parameters

#### **-DriverPackage<DriverPackage>**

Specifies a driver package object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

---

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Tag<List<String>>**

Specifies a word or phrase to associate with an object so that you can search for all objects with the specified set of tags. You can search for a subset of tags, or you can search for the full set of tags.



|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **DriverPackage[]**
- **DriverPackage**

## **Examples**

### **1: Update the tags for a set of driver packages.**

The first command gets all driver package objects that have a tag value of Production and stores the objects in the \$Drivers object array.

The second command uses the ForEach statement to add the value "NewTag" to each driver package object stored in \$Drivers.

---

For information about the Windows PowerShell ForEach statement, type: "Get-Help about\_Foreach".

```
PS C:\> $Drivers = Get-SCDriverPackage -Tag "Production"
```

```
PS C:\> ForEach ($Driver in $Drivers) {Set-SCDriverPackage -DriverPackage $Driver -Tag  
{Production, NewTag}}
```

## Related topics

[Get-SCDriverPackage](#)

[Remove-SCDriverPackage](#)

---

# Set-SCDynamicOptimizationConfiguration

---

## Set-SCDynamicOptimizationConfiguration

Configures dynamic optimization for a host group.

### Syntax

Parameter Set: FromValues

```
Set-SCDynamicOptimizationConfiguration -DynamicOptimizationConfiguration  
<HostGroupDOSettings> [-Aggressiveness <Byte> ] [-EnablePowerOptimization <Boolean> ] [-  
FrequencyMinutes <UInt32> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-  
RunAsynchronously] [-VMMServer <ServerConnection> ] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: ToAutomatic

```
Set-SCDynamicOptimizationConfiguration -AutomaticMode -DynamicOptimizationConfiguration  
<HostGroupDOSettings> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-  
VMMServer <ServerConnection> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ToInherit

```
Set-SCDynamicOptimizationConfiguration -DynamicOptimizationConfiguration  
<HostGroupDOSettings> -Inherit <Boolean> [-JobVariable <String> ] [-PROTipID <Guid> ] [-  
RunAsynchronously] [-VMMServer <ServerConnection> ] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: ToManual

```
Set-SCDynamicOptimizationConfiguration -DynamicOptimizationConfiguration  
<HostGroupDOSettings> -ManualMode [-JobVariable <String> ] [-PROTipID <Guid> ] [-  
RunAsynchronously] [-VMMServer <ServerConnection> ] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

### Detailed Description

The Set-SCDynamicOptimizationConfiguration cmdlet configures dynamic optimization for a host group.

### Parameters

#### **-Aggressiveness<Byte>**

Specifies the level of improvement required before migrating a virtual machine from one host to another in order to load balance virtual machines. The higher the aggressiveness, the more resulting live

---

migrations; the lower the aggressiveness, the fewer resulting live migrations. Valid values are: 1 through 5. The default value is 3 (Medium).

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-AutomaticMode**

Indicates that dynamic optimization automatically migrates virtual machines in order to load balance.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-DynamicOptimizationConfiguration<HostGroupDOSettings>**

Specifies a dynamic optimization configuration object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

---

## **-EnablePowerOptimization<Boolean>**

Enables power optimization when set to \$True.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-FrequencyMinutes<UInt32>**

Specifies the frequency, in minutes, at which dynamic optimization will run when set to automatic mode.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-Inherit<Boolean>**

Indicates whether settings are inherited from the parent host group.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-ManualMode**

Indicates that dynamic optimization will not run automatically.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-Confirm**

Prompts you for confirmation before executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## -WhatIf

Describes what would happen if you executed the command without actually executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **DynamicOptimizationConfiguration**

## Examples

### 1: Enable automatic mode for a dynamic optimization configuration.

The first command gets the host group object named HostGroup01 and stores the object in the \$HostGroup variable.

The second command gets the dynamic optimization configuration object for the host group stored in \$HostGroup and stores the object in the \$DOConfig variable.

The last command enables automatic mode for the dynamic optimization configuration stored in \$DOConfig.

```
PS C:\> $HostGroup = Get-SCVMHostGroup "HostGroup01"
```

```
PS C:\> $DOConfig = Get-SCDynamicOptimizationConfiguration -VMHostGroup $HostGroup
```

```
PS C:\> Set-SCDynamicOptimizationConfiguration -DynamicOptimizationConfiguration $DOConfig -AutomaticMode
```

### 2: Enable power optimization for a host group.

The first command gets the host group object named HostGroup01 and stores the object in the \$HostGroup variable.



---

The second command gets the dynamic optimization configuration object for the host group stored in \$HostGroup and stores the object in the \$DOConfig variable.

The last command enables power optimization for the dynamic optimization configuration stored in \$DOConfig.

```
PS C:\> $HostGroup = Get-SCVMHostGroup "HostGroup01"
```

```
PS C:\> $DOConfig = Get-SCDynamicOptimizationConfiguration -VMHostGroup $HostGroup
```

```
PS C:\> Set-SCDynamicOptimizationConfiguration -DynamicOptimizationConfiguration $DOConfig -  
EnablePowerOptimization $True
```

## Related topics

[Get-SCDynamicOptimizationConfiguration](#)

[Get-SCVMHostGroup](#)

---

# Set-SCExternalJob

---

## Set-SCExternalJob

Updates an existing external job.

### Syntax

Parameter Set: Progress

```
Set-SCExternalJob [-Job] <Task> -ProgressValue <Int32> [ <CommonParameters>]
```

Parameter Set: Completed

```
Set-SCExternalJob [-Job] <Task> -Completed [-InfoMessage <String> ] [ <CommonParameters>]
```

Parameter Set: Failed

```
Set-SCExternalJob [-Job] <Task> -Failed -InfoMessage <String> [ <CommonParameters>]
```

### Detailed Description

The Set-SCExternalJob cmdlets updates existing external jobs in Virtual Machine Manager (VMM). You can update the percent complete of the job, or set the status of the job to completed or failed.

For more information about external jobs, see the New-SCExternalJob cmdlet.

### Parameters

#### -Completed

Sets the status of an external job to completed.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-Failed**

Sets the status of an external job failed.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-InfoMessage<String>**

Provides an informational message for external jobs.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-Job<Task>**

Specifies a VMM job object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

---

## -ProgressValue<Int32>

Supplies the percent complete of the progress of an external job.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- Task

## Examples

### 1: Set an external job to 15% complete.

The first command gets the external job object named ExternalJob01 and stores the object in the \$externalJob variable.

The second command sets the progress value for the job stored in \$externalJob to 15.

```
PS C:\> $externalJob = Get-SCJob -Name "ExternalJob01"
```

```
PS C:\> Set-SCExternalJob -Job $externalJob -ProgressValue 15
```

### 2: Set an external job to completed.

The first command gets the external job object named ExternalJob02 and stores the object in the \$externalJob variable.

The second command sets the status of the job stored in \$externalJob to completed, and adds information about the job.

```
PS C:\> $externalJob = Get-SCJob -Name "ExternalJob02"
```

```
PS C:\> Set-SCExternalJob -Job $externalJob -Completed -InfoMessage "Some information about the completed job"
```

---

### 3: Set an external job to failed.

The first command gets the external job object named ExternalJob03 and stores the object in the \$externalJob variable.

The second command sets the status of the job stored in \$externalJob to failed, and adds information about the job.

```
PS C:\> $externalJob = Get-SCJob -Name "ExternalJob03"
```

```
PS C:\> Set-SCEExternalJob -Job $externalJob -Failed -InfoMessage "An error message for the failed job"
```

### Related topics

[New-SCEExternalJob](#)

---

# Set-SCGuestInfo

---

## Set-SCGuestInfo

Sets the value associated with a key for a key/value pair in a guest operating system.

### Syntax

Parameter Set: MultipleKvpKeys

```
Set-SCGuestInfo [-VM] <VM> -KvpMap <Hashtable> [ <CommonParameters>]
```

Parameter Set: SingleKvpKey

```
Set-SCGuestInfo [-VM] <VM> [-Key] <String> [[-Value] <String> ] [ <CommonParameters>]
```

### Detailed Description

The Set-SCGuestInfo cmdlet sets the value associated with a key for a key/value pair in a guest operating system.

### Parameters

#### **-Key<String>**

Specifies the key in a key/value pair (KVP).

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | 2     |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### **-KvpMap<Hashtable>**

Specifies a hashtable of key/value pairs (KVPs) corresponding to the KVP values exposed by Hyper-V.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Value<String>**

Specifies a string used to attribute an object or property.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | 3     |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VM<VM>**

Specifies a virtual machine object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

---

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **String**

## Examples

### 1: Set a single key/value pair.

The first command gets the virtual machine object named VM01 and stores the object in the \$VM variable.

The second command sets a key/value pair for VM01.

```
PS C:\> $VM = Get-SCVirtualMachine "VM01"
PS C:\> Set-SCGuestInfo -VM $VM -Key Key -Value Value
```

### 2: Set a key to a value for a key/value pair.

This command sets the key to Microsoft.Lab.Isolation.ServerVersion and the value to 1.0.1101 for the virtual machine named VM01. If the key does not exist, it will be created with the specified value. If the key already exists, its value will be overwritten using the value specified in this command.

You can use the Read-SCGuestInfo cmdlet to provide the key and return its corresponding value.

```
PS C:\> Get-SCVirtualMachine -Name "VM01" | Set-SCGuestInfo -Key
Microsoft.Lab.Isolation.ServerVersion -Value 1.0.1101
```

### 3: Set multiple key/value pairs.

The first command gets the virtual machine object named Win2k8R2 and stores the object in the \$VM variable.

The second command creates a hashtable named \$ValuesMap that contains the corresponding keys and values for the key/value pairs. Values can be set to a string, an empty string, or \$null. Setting a value to \$null deletes the key.

The third command sets the key/value pairs for the virtual machine named Win2k8R2.

The last command reads back the key/value pairs for the virtual machine named Win2k8R2.

```
PS C:\> $VM = Get-SCVirtualMachine -Name "Win2k8R2"
PS C:\> $ValuesMap = @{"Key1" = "avalue1"; "Key2IsEmptyString" = "" ; "Key3" = "value3"}
PS C:\> Set-SCGuestInfo -VM $VM -KvpMap $ValuesMap
PS C:\> Read-SCGuestInfo -VM $VM -KvpMap $ValuesMap | select KvpMap
```



---

## 4: Modify a set of values for a set of key/value pairs.

The first command gets the virtual machine object named Win2k8R2 and stores the object in the \$VM variable.

The second command creates a hashtable named \$ValuesMap that contains the corresponding keys and values for the key/value pairs. Values can be set to a string, an empty string, or \$null. Setting a value to \$null deletes the key.

The third command sets the key/value pairs for the virtual machine named Win2k8R2.

The fourth command reads back the key/value pairs for the virtual machine named Win2k8R2.

The fifth command creates a new hashtable where a specific key is changed to a different value.

The sixth command sets the modified value for the specified key in the hashtable.

The last two commands read back the key/value pairs for the virtual machine named Win2k8R2, including the modified value for the key Key2IsEmptyString.

```
PS C:\> $VM = Get-SCVirtualMachine -Name "Win2k8R2"
PS C:\> $ValuesMap = @{"Key1" = "avalue1"; "Key2IsEmptyString" = "" ; "Key3" = "value3"}
PS C:\> Set-SCGuestInfo -VM $VM -KvpMap $ValuesMap
PS C:\> Read-SCGuestInfo -VM $VM -KvpMap $ValuesMap | select KvpMap
PS C:\> $ValuesMap = @{"Key2IsEmptyString" = "KeyIsNoLongerEmpty"}
PS C:\> Set-SCGuestInfo -VM $VM -KvpMap $ValuesMap
PS C:\> $ValuesMap = @{"Key1" = $null; "Key2IsEmptyString" = $null; "Key3" = $null}
PS C:\> Read-SCGuestInfo -VM $VM -KvpMap $ValuesMap | select KvpMap
```

## 5: Delete a key/value pair using two methods.

The first command gets the virtual machine object named Win2k8R2 and stores the object in the \$VM variable.

The next three commands create two keys and their values and return them to the console for virtual machine Win2k8R2.

The fifth command deletes the key/value pair Key1 by calling Set-SCGuestInfo without specifying the value parameter.

The sixth and seventh commands create a new Hashtable with Null as the value for key Key2. Then, key Key2 is deleted by calling the Set-SCGuestInfo command.

The last command shows that both keys that were initially created are now deleted via two separate methods.

```
PS C:\> $VM = Get-SCVirtualMachine -Name "Win2k8R2"
PS C:\> $ValuesMap = @{"Key1" = "avalue1"; "Key2" = "avalue2"}
PS C:\> Set-SCGuestInfo -VM $VM -KvpMap $ValuesMap
PS C:\> Read-SCGuestInfo -VM $VM -KvpMap $ValuesMap | select KvpMap
PS C:\> Set-SCGuestInfo -VM $VM -Key Key1
PS C:\> $KvpsToDelete = @{"Key2" = $null}
PS C:\> Set-SCGuestInfo -VM $VM -KvpMap $KvpsToDelete
PS C:\> Read-SCGuestInfo -VM $VM -KvpMap $ValuesMap | select KvpMap
```

---

## 6: Set multiple values where one value is empty.

The first command gets the virtual machine object named VM01 and stores the object in the \$VM variable.

The second command creates a hashtable named \$ValuesMap3 that contains the corresponding keys and values for the key/value pairs. Values can be set to a string, an empty string, or \$null. Setting a value to \$null deletes the key.

The third command sets the values for the specified keys in the hashtable.

The last command reads back the key/value pairs for the virtual machine named VM01.

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM01"
PS C:\> $ValuesMap3 = @{"VSLM1" = "value1"; "VLSM2" = "value2" ; "VLSM3" = "value3" ;
"VLDM4" = ""}
PS C:\> Set-SCGuestInfo -VM $VM -KVPMAP $ValuesMap3
PS C:\> Read-SCGuestInfo -VM $VM -KVPMAP $ValuesMap3 | select KVPMAP
```

## 7: Delete one value and set another value to empty.

The first command gets the virtual machine object named VM01 and stores the object in the \$VM variable.

The second command creates a hashtable named \$ValuesMap4 that contains the corresponding keys and values for the key/value pairs. Values can be set to a string, an empty string, or \$null. Setting a value to \$null deletes the key.

The third command sets the values for the specified keys in the hashtable.

The fourth command deletes key VLSM2 and sets key VSLM1 to empty by calling the Set-SCGuestInfo command.

The last command reads back the key/value pairs for the virtual machine named VM01.

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM01"
PS C:\> $ValuesMap4 = @{"VLSM2" = $null; "VSLM1" = "" }
PS C:\> Set-SCGuestInfo -VM $VM -KVPMAP $ValuesMap4
PS C:\> Read-SCGuestInfo -VM $VM -KVPMAP $ValuesMap4 | select KVPMAP
```

## 8: Set one value and delete another value.

The first command gets the virtual machine object named VM01 and stores the object in the \$VM variable.

The second command creates a hashtable named \$ValuesMap5 that contains the corresponding keys and values for the key/value pairs. Values can be set to a string, an empty string, or \$null. Setting a value to \$null deletes the key.

The fourth command sets key VSLM1 to "data again" and deletes key VLSM3 by calling the Set-SCGuestInfo command.

---

The last command reads back the key/value pairs for the virtual machine named VM01.

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM01"
PS C:\> $ValuesMap5 = @{"VSLM1" = "data again"; "VLSM3" = $null }
PS C:\> Set-SCGuestInfo -VM $VM -KVPMMap $ValuesMap5
PS C:\> Read-SCGuestInfo -VM $VM -KVPMMap $ValuesMap5 | select KVPMMap
```

## 9: Ignore the deletion of keys that do not exist.

The first command gets the virtual machine object named VM01 and stores the object in the \$VM variable.

The second command creates a hashtable named \$ValuesMap5 that contains the corresponding keys and values for the key/value pairs. Setting a value to \$null deletes the key.

The third command sets the values to \$null for the specified keys in the hashtable.

The last command deletes all keys in the hashtable except for key o1ff1 by calling the Set-SCGuestInfo cmdlet.

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM01"
PS C:\> $KeysDoNotExist = @{"o1ff1" = $null; "o1ff2" = $null; "o1ff3" = $null ; "o1ff4" = $null }
PS C:\> Set-SCGuestInfo -VM $VM -KVPMMap $KeysDoNotExist
PS C:\> Set-SCGuestInfo -VM $VM -Key "o1ff1"
```

## Related topics

[Read-SCGuestInfo](#)

---

# Set-SCGuestOSProfile

---

## Set-SCGuestOSProfile

Changes the properties of a guest operating system profile used in VMM.

### Syntax

Parameter Set: Default

```
Set-SCGuestOSProfile [-GuestOSProfile] <GuestOSProfile> [-AnswerFile <Script> ] [-AutoLogonCount <UInt32> ] [-AutoLogonCredential <RunAsAccount> ] [-ComputerName <String> ] [-Description <String> ] [-DisableAutoLogon] [-DomainJoinOrganizationalUnit <String> ] [-FullName <String> ] [-GuiRunOnceCommands <String[]> ] [-JobVariable <String> ] [-LinuxAdministratorSSHKey <SSHKey> ] [-LinuxDomainName <String> ] [-LocalAdministratorCredential <VMMSecret> ] [-MergeAnswerFile <Boolean> ] [-Name <String> ] [-OperatingSystem <OperatingSystem> ] [-OrganizationName <String> ] [-Owner <String> ] [-ProductKey <String> ] [-PROTipID <Guid> ] [-RemoveDomainJoinOrganizationalUnit] [-RemoveServerFeatures] [-RunAsynchronously] [-TimeZone <Int32> ] [-UnattendSettings <Dictionary<String>> ] [-UserRole <UserRole> ] [ <CommonParameters>]
```

Parameter Set: Domain

```
Set-SCGuestOSProfile [-GuestOSProfile] <GuestOSProfile> -Domain <String> [-AnswerFile <Script> ] [-AutoLogonCount <UInt32> ] [-AutoLogonCredential <RunAsAccount> ] [-ComputerName <String> ] [-Description <String> ] [-DisableAutoLogon] [-DomainJoinCredential <VMMSecret> ] [-DomainJoinOrganizationalUnit <String> ] [-FullName <String> ] [-GuiRunOnceCommands <String[]> ] [-JobVariable <String> ] [-LinuxAdministratorSSHKey <SSHKey> ] [-LinuxDomainName <String> ] [-LocalAdministratorCredential <VMMSecret> ] [-MergeAnswerFile <Boolean> ] [-Name <String> ] [-OperatingSystem <OperatingSystem> ] [-OrganizationName <String> ] [-Owner <String> ] [-ProductKey <String> ] [-PROTipID <Guid> ] [-RemoveDomainJoinOrganizationalUnit] [-RemoveServerFeatures] [-RunAsynchronously] [-TimeZone <Int32> ] [-UnattendSettings <Dictionary<String>> ] [-UserRole <UserRole> ] [ <CommonParameters>]
```

Parameter Set: Workgroup

```
Set-SCGuestOSProfile [-GuestOSProfile] <GuestOSProfile> [-AnswerFile <Script> ] [-AutoLogonCount <UInt32> ] [-AutoLogonCredential <RunAsAccount> ] [-ComputerName <String> ] [-Description <String> ] [-DisableAutoLogon] [-DomainJoinOrganizationalUnit <String> ] [-FullName <String> ] [-GuiRunOnceCommands <String[]> ] [-JobVariable <String> ] [-LinuxAdministratorSSHKey <SSHKey> ] [-LinuxDomainName <String> ] [-LocalAdministratorCredential <VMMSecret> ] [-MergeAnswerFile <Boolean> ] [-Name <String> ] [-OperatingSystem <OperatingSystem> ] [-OrganizationName <String> ] [-Owner <String> ] [-ProductKey <String> ] [-PROTipID <Guid> ] [-RemoveDomainJoinOrganizationalUnit] [-RemoveServerFeatures] [-RunAsynchronously] [-TimeZone <Int32> ] [-UnattendSettings <Dictionary<String>> ] [-UserRole <UserRole> ] [-Workgroup <String> ] [ <CommonParameters>]
```

---

## Detailed Description

The Set-SCGuestOSProfile changes one or more properties of a guest operating system profile used in a Virtual Machine Manager (VMM) environment. Changes made to a guest operating system profile affect only the guest operating system profile itself. Changes do not affect any existing virtual machines that were previously created by using this profile.

## Parameters

### **-AnswerFile<Script>**

Specifies a script object stored in the VMM library to use as an answer file. The name of the answer file script depends on the operating system that you want to install on a virtual machine:

ANSWER FILE    GUEST OS TO INSTALL ON VM

-----

Sysprep.inf    Windows XP, Windows Server 2000, or Windows Server 2003

Unattend.xml    Windows Vista, Windows 7, or Windows Server 2008

|                             |                |
|-----------------------------|----------------|
| Aliases                     | SysPrepFile    |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-AutoLogonCount<UInt32>**

Specifies the number of times that Windows should automatically log the administrator denoted in the Answer File on to the console session.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-AutoLogonCredential<RunAsAccount>**

Specifies the Run As account contained in the Answer File that Windows uses to log on to the console session when automatic administrator logon is enabled.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-ComputerName<String>**

Specifies the name of a computer that VMM can uniquely identify on your network. Valid formats are: FQDN, IPv4 or IPv6 address, or NetBIOS name.

NOTE: See the examples for a specific cmdlet to determine how that cmdlet specifies the computer name.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-Description<String>**

States a description for the specified object.

|                        |       |
|------------------------|-------|
| Aliases                | none  |
| Required?              | false |
| Position?              | named |
| Default Value          | none  |
| Accept Pipeline Input? | false |

|                             |       |
|-----------------------------|-------|
| Accept Wildcard Characters? | false |
|-----------------------------|-------|

## **-DisableAutoLogon**

Disables automatic administrator logon.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-Domain<String>**

Specifies a fully qualified domain name (FQDN) for an Active Directory domain.

Example format: -Domain "Domain01.Corp.Contoso.com"

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-DomainJoinCredential<VMMSCredential>**

Specifies the user name and password of an account with permission to join a computer to the domain. A limited rights account should be used for joining computers (either physical or virtual) to the domain.

Example format for a PS credential:

```
$DomainJoinCredential = Get-Credential
```

```
-Domain "ThisDomain.Corp.Contoso.com" -DomainJoinCredential $DomainJoinCredential
```

Example format for a Run As account:

```
$DomainJoinCredential = Get-SCRunAsAccount -Name "RunAsAcct01"
```

---

-Domain "ThisDomain.Corp.Contoso.com" -DomainJoinCredential \$DomainJoinCredential

NOTE: You can use the DomainJoinCredential parameter to specify credentials (on a VMHostProfile) for joining a physical host computer to the domain, or to specify credentials (on a new or existing template, on a new or existing guest operating system profile, or on a new virtual machine) for joining a virtual machine to the domain.

|                             |                      |
|-----------------------------|----------------------|
| Aliases                     | JoinDomainCredential |
| Required?                   | false                |
| Position?                   | named                |
| Default Value               | none                 |
| Accept Pipeline Input?      | false                |
| Accept Wildcard Characters? | false                |

### **-DomainJoinOrganizationalUnit<String>**

Specifies the organizational unit to which the computer is joined during an unattended mini-setup.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-FullName<String>**

Specifies the name of the person in whose name a virtual machine is registered.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |



---

## **-GuestOSProfile<GuestOSProfile>**

Specifies a guest operating system profile object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-GuiRunOnceCommands<String[]>**

Specifies one or more commands to add to the [GuiRunOnce] section of an unattended answer file (such as Unattend.xml). Use single quotes around each string enclosed in double quotes.

Example format:

```
-GuiRunOnceCommands "C:\APF\APFPostSysPrepCopy.cmd PARAMS1",  
"C:\APF\APFPostSysPrepCopy.cmd PARAMS1"
```

For information about how Windows PowerShell uses quotes, type: "Get-Help about\_Quoting\_Rules"

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | false |

|                             |       |
|-----------------------------|-------|
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-LinuxAdministratorSSHKey<SSHKey>**

Specifies the public key file for a Linux Ssh Key.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-LinuxDomainName<String>**

Specifies a fully qualified domain name (FQDN) to be used in conjunction with Linux operating system specialization.

Example format: -LinuxDomainName "Domain01.Corp.Contoso.com"

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-LocalAdministratorCredential<VMMSCredential>**

Specifies the user name and password for the Local Administrator account (or Linux root account in the case of a Linux compatible Guest Operating System profile).

---

Specifying credentials on a new or existing template, on a new or existing guest operating system profile, or on a new virtual machine overrides any existing Administrator password.

|                             |                         |
|-----------------------------|-------------------------|
| Aliases                     | AdminPasswordCredential |
| Required?                   | false                   |
| Position?                   | named                   |
| Default Value               | none                    |
| Accept Pipeline Input?      | false                   |
| Accept Wildcard Characters? | false                   |

### **-MergeAnswerFile<Boolean>**

Specifies that the cmdlet merge the specified answer file with the specified guest operating system settings. The default value is TRUE. This parameter is used by the VMM console. You do not need to use this parameter at the command prompt.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Name<String>**

Specifies the name of a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-OperatingSystem<OperatingSystem>**

Specifies the type of operating system for a virtual machine. To list the names of all available operating systems in VMM, type: "Get-SCOperatingSystem".

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-OrganizationName<String>**

Specifies the name of the organization for the person in whose name a virtual machine is registered.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-Owner<String>**

Specifies the owner of a VMM object in the form of a valid domain user account.

Example format: -Owner "Contoso\ReneeLo"

Example format: -Owner "ReneeLo@Contoso"

|                        |       |
|------------------------|-------|
| Aliases                | none  |
| Required?              | false |
| Position?              | named |
| Default Value          | none  |
| Accept Pipeline Input? | false |

---

|                             |       |
|-----------------------------|-------|
| Accept Wildcard Characters? | false |
|-----------------------------|-------|

### **-ProductKey<String>**

Specifies a product key. The product key is a 25-digit number that identifies the product license. A product key can be used to register VMM or an operating system to be installed on a virtual machine or host.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RemoveDomainJoinOrganizationalUnit**

Removes the organizational unit that the computer joined during setup.

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | false |
| Position? | named |

|                             |       |
|-----------------------------|-------|
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RemoveServerFeatures**

Removes all server roles and features from a template.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-TimeZone<Int32>**

Specifies a number (an index) that identifies a geographical region that shares the same standard time. For a list of time zone indexes, see "Microsoft Time Zone Index Values" at: <http://go.microsoft.com/fwlink/?LinkId=120935>. If no time zone is specified, the default time zone used for a virtual machine is the same time zone setting that is on the virtual machine host.

Example format to specify the GMT Standard Time zone: -TimeZone 085

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-UnattendSettings<Dictionary<String>>**

Specifies a key/value pair consisting of string, string.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-UserRole<UserRole>**

Specifies a user role object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Workgroup<String>**

Specifies on a new or existing template, on a new or existing guest operating system profile, or on a new virtual machine the name of the workgroup to which you want to join a virtual machine. You can use this parameter to override the existing value on a template or on a guest operating system profile.

---

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **GuestOSProfile**

## Notes

- Requires a VMM guest OS profile object, which can be retrieved by using the Get-SCGuestOSProfile cmdlet.

## Examples

### 1: Specify an organization name for an existing guest operating system profile.

The first command gets the guest OS profile object named NewOSProfile01 and stores the object in the \$OSProfile variable.

The second command sets Contoso as the organization name for the guest operating system profile stored in \$OSProfile.

```
PS C:\> $OSProfile = Get-SCGuestOSProfile -Name "NewOSProfile01"
```

```
PS C:\> Set-SCGuestOSProfile -GuestOSProfile $OSProfile -OrgName "Contoso"
```



---

## 2: Specify an SSHKey for an existing Linux guest operating system profile.

The first command gets the guest OS profile object named MyCentOSProfile and stores the object in the \$OSProfile variable.

The second command gets the SSHKey object named My.sshkey and stores the object in the \$sshkey variable

The last command sets the My.sshkey object on the guest operating system profile stored in \$OSProfile.

```
PS C:\> $OSProfile = Get-SCGuestOSProfile -Name "My CentOS Profile"
```

```
PS C:\> $sshkey = Get-SCSSHKey -Name "My.sshkey"
```

```
PS C:\> Set-SCGuestOSProfile -GuestOSProfile $OSProfile -LinuxAdministratorSSHKey $sshkey
```

### Related topics

[Get-SCGuestOSProfile](#)

[New-SCGuestOSProfile](#)

[New-SCVirtualMachine](#)

[New-SCVMTemplate](#)

[Remove-SCGuestOSProfile](#)

[Set-SCVMTemplate](#)

---

# Set-SCHardwareProfile

---

## Set-SCHardwareProfile

Changes the properties of a hardware profile used in VMM.

### Syntax

Parameter Set: Default

```
Set-SCHardwareProfile [-HardwareProfile] <HardwareProfile> [-BootOrder {Floppy | CD |  
IdeHardDrive | PxeBoot | Unknown} ] [-CapabilityProfile <CapabilityProfile> ] [-CPUCount  
<Byte> ] [-CPUExpectedUtilizationPercent <Int32> ] [-CPULimitForMigration <Boolean> ] [-  
CPULimitFunctionality <Boolean> ] [-CPUMaximumPercent <Int32> ] [-  
CPUPerVirtualNumaNodeMaximum <Byte> ] [-CPURelativeWeight <Int32> ] [-CPUReserve <Int32> ]  
[-CPUType <ProcessorType> ] [-Description <String> ] [-DiskIops <Int32> ] [-  
DRProtectionRequired <Boolean> ] [-DynamicMemoryBufferPercentage <Int32> ] [-  
DynamicMemoryEnabled <Boolean> ] [-DynamicMemoryMaximumMB <Int32> ] [-DynamicMemoryMinimumMB  
<UInt16> ] [-HAVMPriority <UInt32> ] [-HighlyAvailable <Boolean> ] [-JobGroup <Guid> ] [-  
JobVariable <String> ] [-MemoryMB <Int32> ] [-MemoryPerVirtualNumaNodeMaximumMB <Int32> ] [-  
MemoryWeight <Int32> ] [-MonitorMaximumCount <Int32> ] [-MonitorMaximumResolution <String> ]  
[-Name <String> ] [-NetworkUtilizationMbps <Int32> ] [-NumaIsolationRequired <Boolean> ] [-  
NumLock <Boolean> ] [-Owner <String> ] [-PROTipID <Guid> ] [-RemoveCapabilityProfile] [-  
RunAsynchronously] [-UserRole <UserRole> ] [-VirtualNumaNodesPerSocketMaximum <Byte> ] [-  
VirtualVideoAdapterEnabled <Boolean> ] [ <CommonParameters>]
```

### Detailed Description

The Set-SCHardwareProfile cmdlet changes one or more properties of a hardware profile object used in a Virtual Machine Manager (VMM) environment. Properties that you can change include settings for boot order, CPU settings, the amount memory on the host that is assigned to a virtual machine, and other options.

To change the properties of a virtual floppy drive, virtual DVD drive, virtual COM port, virtual network adapter, or virtual SCSI adapter associated with a specific hardware profile, use the Set-SCVirtualFloppyDrive, Set-SCVirtualDVDDrive, Set-SCVirtualCOMPort, Set-SCVirtualNetworkAdapter, or Set-SCVirtualScsiAdapter cmdlets, respectively.

Changes made to a hardware profile affect only the hardware profile itself. The changes do not affect any existing virtual machines that were created by using this profile.

---

## Parameters

### **-BootOrder<BootDevice[]>**

Specifies the order of devices that a virtual machine on a Hyper-V host uses to start up. Valid values are: CD, IDEHardDrive, PXEBoot, Floppy.

Example format: -BootOrder PXEBoot,IDEHardDrive,CD,Floppy

The acceptable values for this parameter are:

|              |  |
|--------------|--|
| Floppy       |  |
| CD           |  |
| IdeHardDrive |  |
| PxeBoot      |  |
| Unknown      |  |

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-CapabilityProfile<CapabilityProfile>**

Specifies a capability profile object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-CPUCount<Byte>**

Specifies the number of CPUs on a virtual machine, on a hardware profile, or on a template. See the examples for a specific cmdlet to determine how that cmdlet uses this parameter.

TYPE OF HOST    NUMBER OF PROCESSORS

-----    -----

Hyper-V        Up to 4 CPUs per VM; varies by guest OS

VMware ESX    Up to 4 CPUs per VM for any supported guest OS

Exception: 1 CPU on a VM running Windows NT 4.0

Citrix XenServer Up to 8 CPUs per VM; varies by guest OS

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-CPUExpectedUtilizationPercent<Int32>**

Specifies the percent of CPU on the host that you expect this virtual machine to use. This value is used only when VMM determines a suitable host for the virtual machine.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-CPULimitForMigration<Boolean>**

Limits, when set to \$True, processor features for the specified virtual machine in order to enable migration to a physical computer that has a different version of the same processor as the source computer. VMM does not support migrating virtual machines between physical computers that have processors from different manufacturers.

|                             |                      |
|-----------------------------|----------------------|
| Aliases                     | LimitCPUForMigration |
| Required?                   | false                |
| Position?                   | named                |
| Default Value               | none                 |
| Accept Pipeline Input?      | false                |
| Accept Wildcard Characters? | false                |

### **-CPULimitFunctionality<Boolean>**

Enables running an older operating system (such as Windows NT 4.0) on a virtual machine deployed on a Hyper-V host or on a VMware ESX host by providing limited CPU functionality for the virtual machine.

|                             |                       |
|-----------------------------|-----------------------|
| Aliases                     | LimitCPUFunctionality |
| Required?                   | false                 |
| Position?                   | named                 |
| Default Value               | none                  |
| Accept Pipeline Input?      | false                 |
| Accept Wildcard Characters? | false                 |

### **-CPUMaximumPercent<Int32>**

Specifies the highest percentage of the total resources of a single CPU on the host that can be used by a specific virtual machine at any given time.

Example: -CPUMaximumPercent 80 (to specify 80 per cent)

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-CPUPerVirtualNumaNodeMaximum<Byte>**

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-CPURelativeWeight<Int32>**

Specifies the amount of CPU resources on a host that this virtual machine can use relative to other virtual machines on the same host. A virtual machine with a higher setting is allocated more CPU resources than a virtual machine with a lower setting.

TYPE OF HOST    RANGE OF RELATIVE VALUES

-----    -----

Hyper-V        1 to 10000

VMware ESX    2000 = High

1500 = Above Normal

1000 = Normal (default)

750 = Below Normal

500 = Low

1 to 1000000 = Custom

The VMware term for these values is "shares."

Citrix XenServer 1 to 65536, normal is 256

NOTE: See the examples for a specific cmdlet to determine how that cmdlet uses this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-CPUReserve<Int32>**

Specifies the minimum percentage of the resources of a single CPU on the host to allocate to a virtual machine. The percentage of CPU capacity that is available to the virtual machine is never less than this percentage.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-CPUType<ProcessorType>**

Specifies the type of CPU for a virtual machine. To retrieve a list of all CPU types that are available for use in virtual machines in a VMM environment, type: "Get-SCCPUType"

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-Description<String>**

States a description for the specified object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-DiskIops<Int32>**

Specifies the number of disk input/output operations per second (IOPS) on the host that can be used by a specific virtual machine.

Example: -DiskIO 1500 (to specify 1500 IOPS).

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-DRProtectionRequired<Boolean>**

This parameter is reserved for future use.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-DynamicMemoryBufferPercentage<Int32>**

Specifies the percentage of memory above a virtual machine's current memory allocation which the host should try to reserve as a buffer. The default value is 20.

Example format: -DynamicMemoryTargetBufferPercentage 20

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | false |
| Position? | named |



|                             |       |
|-----------------------------|-------|
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-DynamicMemoryEnabled<Boolean>**

Enables, when set to \$True, dynamic memory for virtual machines. You can enable dynamic memory directly on a virtual machine, or on a template or hardware profile that will be used to create virtual machines. The default value is False.

REQUIRED: You can enable dynamic memory for a virtual machine only if that virtual machine is deployed on a host running Windows Server 2008 SP1 or later or if the virtual machine is stored in a library in a stopped state (hardware changes to a stored virtual machine can only be made if the virtual machine does not have snapshots). Enabling dynamic memory on a virtual machine stored in a library will limit placement of that machine to hosts running Windows Server 2008 SP1 or later.

Example format: -DynamicMemoryEnabled \$True

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-DynamicMemoryMaximumMB<Int32>**

Specifies the maximum amount of memory that can be allocated to a virtual machine if dynamic memory is enabled. The default value is 65536.

REQUIRED: You can enable dynamic memory for a virtual machine only if that virtual machine is deployed on a host running Windows Server 2008 R2 SP1 or later or if the virtual machine is stored in a library in a stopped state (hardware changes to a stored virtual machine can only be made if the virtual machine does not have snapshots). Enabling dynamic memory on a virtual machine stored in a library will limit placement of that machine to hosts running Windows Server 2008 R2 SP1 or later.

Example format: -DynamicMemoryMaximumMB 1024

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | false |
| Position? | named |

---

|                             |       |
|-----------------------------|-------|
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-DynamicMemoryMinimumMB<UInt16>**

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-HardwareProfile<HardwareProfile>**

Specifies a hardware profile object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-HAVMPriority<UInt32>**

|               |       |
|---------------|-------|
| Aliases       | none  |
| Required?     | false |
| Position?     | named |
| Default Value | none  |

---

|                             |       |
|-----------------------------|-------|
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-HighlyAvailable<Boolean>**

Specifies that a virtual machine will be placed on a Hyper-V host that is part of a host cluster. Configure this setting on a virtual machine, or on a template or hardware profile that will be used to create virtual machines.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobGroup<Guid>**

Specifies an identifier for a series of commands that will run as a set just before the final command that includes the same job group identifier runs.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | false |

|                             |       |
|-----------------------------|-------|
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-MemoryMB<Int32>**

Specifies, in megabytes (MB), the amount of random access memory (RAM) on the host that is allocated to a virtual machine. The default value is 512 MB. For a virtual machine on which dynamic memory is enabled (on a host running Windows Server 2008 R2 SP1 or later), use MemoryMB to specify the startup memory value.

TYPE OF HOST                      MAXIMUM HOST MEMORY ASSIGNABLE TO VM

-----                      -----

Hyper-V                      Up to 65536 MB RAM per virtual machine

VMware ESX Server 3.0.x Up to 16384 MB RAM per virtual machine

VMware ESX Server 3.5.x Up to 65532 MB RAM per virtual machine

Citrix XenServer    Up to 32265 MB RAM per VM

Example format: -MemoryMB 1024

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-MemoryPerVirtualNumaNodeMaximumMB<Int32>**

|                        |       |
|------------------------|-------|
| Aliases                | none  |
| Required?              | false |
| Position?              | named |
| Default Value          | none  |
| Accept Pipeline Input? | false |

|                             |       |
|-----------------------------|-------|
| Accept Wildcard Characters? | false |
|-----------------------------|-------|

### **-MemoryWeight<Int32>**

Indicates the priority in allocating memory to a virtual machine, relative to other virtual machines on the same host. A virtual machine with a higher setting is allocated more memory resources than a virtual machine with a lower setting.

For a host running Windows Server 2008 R2 SP1 or later, 5000 = Normal, 10000 = High, 0 = Low, 1 to 10000 = Custom.

NOTE: See the examples for a specific cmdlet to determine how that cmdlet uses this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-MonitorMaximumCount<Int32>**

Specifies the maximum number of monitors supported by a virtual video adapter.

Example format: -MonitorMaximumCount 3

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-MonitorMaximumResolution<String>**

Specifies, as a string, the value that represents the maximum possible monitor resolution of a virtual video adapter. Valid values are: "1024x768", "1280x1024", "1600x1200", "1920x1200". Default value: "1280x1024"

---

Example format: -MonitorResolutionMaximum "1600x1200"

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Name<String>**

Specifies the name of a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-NetworkUtilizationMbps<Int32>**

Specifies, in megabits per second (Mbps), the amount of bandwidth on the host's network that can be used by a specific virtual machine.

Example format: -NetworkUtilization 10

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-NumalsolationRequired<Boolean>**

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-NumLock<Boolean>**

Enables the BIOS value for NumLock on a virtual machine (or on a template or hardware profile that is used to create virtual machines) on a Hyper-V host. This parameter does not apply to virtual machines on VMware ESX hosts, or on Citrix XenServer hosts.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-Owner<String>**

Specifies the owner of a VMM object in the form of a valid domain user account.

Example format: -Owner "Contoso\ReneeLo"

Example format: -Owner "ReneeLo@Contoso"

|                        |       |
|------------------------|-------|
| Aliases                | none  |
| Required?              | false |
| Position?              | named |
| Default Value          | none  |
| Accept Pipeline Input? | false |

---

|                             |       |
|-----------------------------|-------|
| Accept Wildcard Characters? | false |
|-----------------------------|-------|

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RemoveCapabilityProfile**

Removes one or more specified capability profile objects.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                        |       |
|------------------------|-------|
| Aliases                | none  |
| Required?              | false |
| Position?              | named |
| Default Value          | none  |
| Accept Pipeline Input? | false |



|                             |       |
|-----------------------------|-------|
| Accept Wildcard Characters? | false |
|-----------------------------|-------|

### **-UserRole<UserRole>**

Specifies a user role object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VirtualNumaNodesPerSocketMaximum<Byte>**

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VirtualVideoAdapterEnabled<Boolean>**

Enables, when set to \$True, the Microsoft Synthetic 3D Virtual Video Adapter for virtual machines. You can enable the Virtual Video Adapter directly on a virtual machine, or on a template or hardware profile that will be used to create virtual machines.

REQUIRED: You can enable the Microsoft Synthetic 3D Virtual Video Adapter for a virtual machine only if that virtual machine is deployed on a host running Windows Server 2008 R2 SP1 (with the Remote Desktop Services role and Remote Desktop Virtual Graphics role service installed) or later or if the virtual machine is stored in a library in a stopped state (hardware changes to a stored virtual machine can only be made if the virtual machine does not have snapshots). Enabling the Microsoft Synthetic 3D Virtual Video Adapter on a virtual machine stored in a library will limit placement of that machine to hosts running Windows Server 2008 R2 SP1 (with the Remote Desktop Services role and Remote Desktop Virtual Graphics role service installed) or later.

---

Example format: -VirtualVideoAdapterEnabled \$TRUE

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **HardwareProfile**

## Examples

### 1: Specify an amount of memory for an existing hardware profile.

The first command gets the hardware profile object named NewHWProfile01 and stores the object in the \$HWProfile variable.

The second command changes the memory value for NewHWProfile01 to 1024 MB.

```
PS C:\> $HWProfile = Get-SCHardwareProfile | where { $_.Name -eq "NewHWProfile01" }
PS C:\> Set-SCHardwareProfile -HardwareProfile $HWProfile -MemoryMB 1024
```

### 2: Specify a new owner for multiple hardware profiles.

The first command gets the hardware profile objects that match the search criteria and stores the objects in the \$HWProfiles object array.

The second command uses a foreach statement to specify a new owner for each of the profiles in the array.

For more information about the standard Windows PowerShell foreach loop statement, type: Get-Help about\_ForEach.

```
PS C:\> $HWProfiles = Get-SCHardwareProfile | where { $_.Name -match "Profile" }
```

---

```
PS C:\> Foreach ($HWProfile in $HWProfiles) {Set-SCHardwareProfile -HardwareProfile $HWProfile -Owner "Contoso\Cesar"}
```

### 3: Specify a new boot order for multiple hardware profiles.

The first command gets all hardware profile objects the library that match the search criteria (the profile name contains the string "HWProfile") and stores the hardware profile objects in the \$HWProfiles object array. Using the "@" symbol and parentheses ensures that the command stores the results in an array, in case the command returns a single object or a null value.

The second command uses a foreach statement to specify a new boot order for each hardware profile object in the \$HWProfiles array.

```
PS C:\> $HWProfiles = @(Get-SCHardwareProfile | where {$_.Name -match "HWProfile"})
PS C:\> Foreach ($HWProfile in $HWProfiles) {Set-SCHardwareProfile -HardwareProfile $HWProfile -BootOrder PXEBoot,CD,IDEHardDrive,Floppy}
```

### 4: Search for hardware profiles with a specific configuration and append text to the description field.

The first command gets all hardware profile objects that match the search criteria (CPU Count is equal to 4) and stores the hardware profile objects in the \$HWProfiles object array.

The second command uses a foreach statement to iterate through each profile object in the \$HWProfiles array. For each profile, the Description text is stored to a variable (\$Text), and then the Set-SCHardwareProfile cmdlet uses the Description parameter to append "(Contains 4 Processors)" to the contents of each instance of \$Text.

```
PS C:\> $HWProfiles = @(Get-SCHardwareProfile | where {$_.CPUCount -eq 4})
PS C:\> Foreach ($HWProfile in $HWProfiles) {$Text = $HWProfile.Description; Set-SCHardwareProfile -HardwareProfile $HWProfile -Description "$Text" (Contains 4 Processors)}
```

### 5: Enable Dynamic Memory for an existing hardware profile.

The first command gets the hardware profile object named NewProfile5 and stores the object in the \$HWProfile variable.

The second command enables Dynamic Memory for NewHWProfile05, changes the startup memory value to 1024 MB and sets the maximum memory value to 2048 MB.

```
PS C:\> $HWProfile = Get-SCHardwareProfile | where { $_.Name -eq "NewHWProfile05" }
PS C:\> Set-SCHardwareProfile -HardwareProfile $HWProfile -DynamicMemoryEnabled $True -MemoryMB 1024 -DynamicMemoryMaximumMB 2048
```

## Related topics

[Get-SCHardwareProfile](#)

---

[New-SCHardwareProfile](#)

[Set-SCVirtualCOMPort](#)

[Set-SCVirtualDVDDrive](#)

[Set-SCVirtualFloppyDrive](#)

[Set-SCVirtualNetworkAdapter](#)

[Set-SCVirtualScsiAdapter](#)

---

# Set-SCHostReserve

---

## Set-SCHostReserve

Modifies the host reserve settings for a host group.

### Syntax

Parameter Set: FromCPUEnabled

```
Set-SCHostReserve -CPU -Enabled <Boolean> -HostReserve <HostReserveSettings> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: FromCPUPlacementLevel

```
Set-SCHostReserve -CPU -HostReserve <HostReserveSettings> -PlacementLevel <UInt64> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: FromCPUStartOptimizationLevel

```
Set-SCHostReserve -CPU -HostReserve <HostReserveSettings> -StartOptimizationLevel <UInt64> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: FromCPUVMHostReserveLevel

```
Set-SCHostReserve -CPU -HostReserve <HostReserveSettings> -VMHostReserveLevel <UInt64> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: FromDiskIOEnabled

```
Set-SCHostReserve -DiskIO -Enabled <Boolean> -HostReserve <HostReserveSettings> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: FromDiskIOIOPS

```
Set-SCHostReserve -DiskIO -HostReserve <HostReserveSettings> -IOPS [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: FromDiskIOPercent

```
Set-SCHostReserve -DiskIO -HostReserve <HostReserveSettings> -Percent [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: FromDiskIOPlacementLevel

```
Set-SCHostReserve -DiskIO -HostReserve <HostReserveSettings> -PlacementLevel <UInt64> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

---

Parameter Set: FromDiskIOStartOptimizationLevel

Set-SCHostReserve -DiskIO -HostReserve <HostReserveSettings> -StartOptimizationLevel <UInt64> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [-Confirm] [-WhatIf] [ <CommonParameters>]

Parameter Set: FromDiskIOVMHostReserveLevel

Set-SCHostReserve -DiskIO -HostReserve <HostReserveSettings> -VMHostReserveLevel <UInt64> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [-Confirm] [-WhatIf] [ <CommonParameters>]

Parameter Set: FromDiskSpaceEnabled

Set-SCHostReserve -DiskSpace -Enabled <Boolean> -HostReserve <HostReserveSettings> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [-Confirm] [-WhatIf] [ <CommonParameters>]

Parameter Set: FromDiskSpaceGB

Set-SCHostReserve -DiskSpace -GB -HostReserve <HostReserveSettings> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [-Confirm] [-WhatIf] [ <CommonParameters>]

Parameter Set: FromDiskSpaceMB

Set-SCHostReserve -DiskSpace -HostReserve <HostReserveSettings> -MB [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [-Confirm] [-WhatIf] [ <CommonParameters>]

Parameter Set: FromDiskSpacePercent

Set-SCHostReserve -DiskSpace -HostReserve <HostReserveSettings> -Percent [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [-Confirm] [-WhatIf] [ <CommonParameters>]

Parameter Set: FromDiskSpacePlacementLevel

Set-SCHostReserve -DiskSpace -HostReserve <HostReserveSettings> -PlacementLevel <UInt64> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [-Confirm] [-WhatIf] [ <CommonParameters>]

Parameter Set: FromDiskSpaceVMHostReserveLevel

Set-SCHostReserve -DiskSpace -HostReserve <HostReserveSettings> -VMHostReserveLevel <UInt64> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [-Confirm] [-WhatIf] [ <CommonParameters>]

Parameter Set: FromMemoryEnabled

Set-SCHostReserve -Enabled <Boolean> -HostReserve <HostReserveSettings> -Memory [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [-Confirm] [-WhatIf] [ <CommonParameters>]

Parameter Set: FromMemoryGB

Set-SCHostReserve -GB -HostReserve <HostReserveSettings> -Memory [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [-Confirm] [-WhatIf] [ <CommonParameters>]

Parameter Set: FromMemoryMB

Set-SCHostReserve -HostReserve <HostReserveSettings> -MB -Memory [-JobVariable <String> ] [-

---

PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [-Confirm] [-WhatIf]  
[ <CommonParameters>]

Parameter Set: FromMemoryPercent

Set-SCHostReserve -HostReserve <HostReserveSettings> -Memory -Percent [-JobVariable <String>  
] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [-Confirm] [-  
WhatIf] [ <CommonParameters>]

Parameter Set: FromMemoryPlacementLevel

Set-SCHostReserve -HostReserve <HostReserveSettings> -Memory -PlacementLevel <UInt64> [-  
JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer  
<ServerConnection> ] [-Confirm] [-WhatIf] [ <CommonParameters>]

Parameter Set: FromMemoryStartOptimizationLevel

Set-SCHostReserve -HostReserve <HostReserveSettings> -Memory -StartOptimizationLevel  
<UInt64> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer  
<ServerConnection> ] [-Confirm] [-WhatIf] [ <CommonParameters>]

Parameter Set: FromMemoryVMHostReserveLevel

Set-SCHostReserve -HostReserve <HostReserveSettings> -Memory -VMHostReserveLevel <UInt64> [-  
JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer  
<ServerConnection> ] [-Confirm] [-WhatIf] [ <CommonParameters>]

Parameter Set: FromNetworkEnabled

Set-SCHostReserve -Enabled <Boolean> -HostReserve <HostReserveSettings> -Network [-  
JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer  
<ServerConnection> ] [-Confirm] [-WhatIf] [ <CommonParameters>]

Parameter Set: FromNetworkMbps

Set-SCHostReserve -HostReserve <HostReserveSettings> -Mbps -Network [-JobVariable <String> ]  
[-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [-Confirm] [-  
WhatIf] [ <CommonParameters>]

Parameter Set: FromNetworkPercent

Set-SCHostReserve -HostReserve <HostReserveSettings> -Network -Percent [-JobVariable  
<String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [-  
Confirm] [-WhatIf] [ <CommonParameters>]

Parameter Set: FromNetworkPlacementLevel

Set-SCHostReserve -HostReserve <HostReserveSettings> -Network -PlacementLevel <UInt64> [-  
JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer  
<ServerConnection> ] [-Confirm] [-WhatIf] [ <CommonParameters>]

Parameter Set: FromNetworkStartOptimizationLevel

Set-SCHostReserve -HostReserve <HostReserveSettings> -Network -StartOptimizationLevel  
<UInt64> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer  
<ServerConnection> ] [-Confirm] [-WhatIf] [ <CommonParameters>]

Parameter Set: FromNetworkVMHostReserveLevel

Set-SCHostReserve -HostReserve <HostReserveSettings> -Network -VMHostReserveLevel <UInt64>  
[-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer  
<ServerConnection> ] [-Confirm] [-WhatIf] [ <CommonParameters>]

---

Parameter Set: ToInherit

```
Set-SCHostReserve -HostReserve <HostReserveSettings> -Inherit <Boolean> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

## Detailed Description

The Set-SCHostReserve cmdlet modifies the host reserve settings for a host group. To modify the host reserve settings for a host group, that host group must not be inheriting its settings from a parent host group.

When you set the host reserve levels, the unit parameters, such as GB or Percentage, dictate the units in which the other levels, such as StartOptimizationLevel and PlacementLevel, are expressed.

The values for VMHostReserveLevel, StartOptimizationLevel and PlacementLevel must be represented in order. For example, the value for StartOptimizationLevel cannot be less than the value for VMHostReserveLevel, and the value for PlacementLevel cannot be less than the value for StartOptimizationLevel.

When the host has less than the specified amount for StartOptimizationLevel available, Dynamic Optimization will automatically try to rebalance the load.

A host will never be forced by Power Optimization to have less than the specified amount for PlacementLevel available because of another node being powered off.

## Parameters

### -CPU

Specifies a host reserve CPU resource type.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### -DiskIO

Specifies a host reserve Disk I/O resource type.



|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-DiskSpace**

Specifies a host reserve disk space resource type.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-Enabled<Boolean>**

Enables an object when set to \$True, or disables an object when set to \$False. For example, if you want to upgrade software on a virtual machine template, you can disable the template object in the VMM library to temporarily prevent users from using that object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-GB**

Indicates that the unit for a host reserve resource is expressed in gigabytes (GB).

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-HostReserve<HostReserveSettings>**

Specifies a host reserve settings object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-Inherit<Boolean>**

Indicates whether settings are inherited from the parent host group.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-IOPS**

Indicates that the unit for a host reserve resource is expressed in disk input/output operations per second (IOPS).

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-MB**

Indicates that the unit for a host reserve resource is expressed in megabytes (MB).

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-Mbps**

Indicates that the unit for a host reserve resource is expressed in megabits per second (Mbps).

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-Memory**

Specifies a host reserve memory resource type.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-Network**

Specifies a host reserve Network I/O resource type.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-Percent**

Indicates that the unit for a host reserve resource is expressed in percent (%).

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-PlacementLevel<UInt64>**

Specifies the host reserve level above which placement is acceptable.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-StartOptimizationLevel<UInt64>**

Specifies the host reserve level at which dynamic optimization is started.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-VMHostReserveLevel<UInt64>**

Specifies the host reserve level at which placement returns an error if starting a virtual machine would require dropping below this level.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

---

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-Confirm**

Prompts you for confirmation before executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **HostReserve**

## Examples

### 1: Modify the CPU host reserve and placement settings for a specified host group.

The first command gets the host group object named HostGroup01 and stores the object in the \$HostGroup variable.

The second command gets the host reserve object for the host group stored in \$HostGroup, and then stores the object in the \$HostReserve variable.

The last command uses the pipeline operator to pass the host reserve stored in \$HostReserve to the Set-SCHostReserve cmdlet which updates the host reerve and placement settings.

```
PS C:\> $HostGroup = Get-SCVMHostGroup "HostGroup01"
```

```
PS C:\> $HostReserve = Get-SCHostReserve -VMHostGroup $HostGroup
```

```
PS C:\> $HostReserve | Set-SCHostReserve -CPU -PlacementLevel 75 -StartOptimizationLevel 80  
-VMHostReserveLevel 90
```

## Related topics

[Get-SCHostReserve](#)

[Get-SCVMHostGroup](#)



---

# Set-SCInternetSCSIHba

---

## Set-SCInternetSCSIHba

Configures iSCSI initiator sessions and logs the initiator on to the iSCSI storage array.

### Syntax

Parameter Set: CreateSession

```
Set-SCInternetSCSIHba -CreateSession -InternetSCSIHba <HostInternetSCSIHba> -TargetPortal <StorageiSCSIPortal> [-InitiatorIP <String> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-TargetName <StorageEndpoint> ] [ <CommonParameters>]
```

Parameter Set: CreateSessionArray

```
Set-SCInternetSCSIHba -CreateSession -InternetSCSIHba <HostInternetSCSIHba> -StorageArray <StorageArray> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

### Detailed Description

The Set-InternetSCSIHBA cmdlet configures the iSCSI initiator on a virtual machine host. Set-InternetSCSIHBA also creates persistent connections to the iSCSI storage array by either explicitly specifying the initiator IP, target name, and target portal or by allowing Virtual Machine Manager (VMM) to automatically create multiple connections to the iSCSI storage array based on subnet matching.

### Parameters

#### -CreateSession

Indicates that an iSCSI session is created.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

### **-InitiatorIP<String>**

Specifies an IP address for the iSCSI initiator.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-InternetSCSIHba<HostInternetSCSIHba>**

Specifies an instance of an iSCSI initiator on a host.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-StorageArray<StorageArray>**

Specifies a storage array object, This can be a Fibre Channel or iSCSI storage sub-system that is used to store virtual machine configuration and virtual disks.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-TargetName<StorageEndpoint>**

Specifies the name for an iSCSI storage array target.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-TargetPortal<StorageiSCSIPortal>**

Specifies an iSCSI storage array target portal object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **HostInternetSCSIHBA**

## **Related topics**

[Get-SCStorageArray](#)

[Get-SCVMHost](#)

---

# Set-SCIPAddress

---

## Set-SCIPAddress

Modifies an allocated IP address by assigning the IP address to an object, or updating the IP address description.

### Syntax

Parameter Set: Default

```
Set-SCIPAddress [-AllocatedIPAddress] <AllocatedIPAddress> [-Description <String> ] [-GrantToObjectID <Guid> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Set-SCIPAddress cmdlet modifies an allocated IP address. You can use Set-SCIP address to assign an allocated IP address to an object using the GrantToObjectID parameter.

### Parameters

#### **-AllocatedIPAddress<AllocatedIPAddress>**

Specifies an IP address that has been allocated from an IP address pool.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

#### **-Description<String>**

States a description for the specified object.

---

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-GrantToObjectID<Guid>**

Specifies the ID of an object to which an allocated IP address or MAC address will be assigned.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

---

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **CloudDRPairingInfoData**

## Examples

### 1: Assign an allocated IP address to a load balancer virtual IP.

The first command gets the load balancer virtual IP object named LoadBalancerVIP01 and stores the object in the \$VIP variable.

The second command gets the static IP address pool object with the specified IPv4 subnet and stores the object in the \$IPAddressPool variable.

The third command gets the unassigned IP address objects for the static IP address pool stored in \$IPAddressPool and stores the objects in the \$IPAddress variable.

The last command assigns the first unassigned IP address from the addresses stored in \$IPAddress to the virtual load balancer ID stored in \$VIP.ID.

```
PS C:\> $VIP = Get-SCLoadBalancerVIP -Name "LoadBalancerVIP01"
```

```
PS C:\> $IPAddressPool = Get-SCStaticIPAddressPool -IPv4 -Subnet "10.0.0.0/24"
```

```
PS C:\> $IPAddress = Get-SCIPAddress -StaticIPAddressPool $IPAddressPool -Unassigned
```

```
PS C:\> Set-SCIPAddress -AllocatedIPAddress $IPAddress[0] -GrantToObjectID $VIP.ID
```

## Related topics

[Get-SCIPAddress](#)

[Grant-SCIPAddress](#)

[Revoke-SCIPAddress](#)



---

# Set-SCISO

---

## Set-SCISO

Changes the properties of an ISO object.

### Syntax

Parameter Set: Default

```
Set-SCISO [-ISO] <ISO> [-Description <String> ] [-Enabled <Boolean> ] [-FamilyName <String> ] [-JobVariable <String> ] [-Name <String> ] [-Owner <String> ] [-PROTipID <Guid> ] [-Release <String> ] [-RunAsynchronously] [-SharePath <String> ] [-UserRole <UserRole> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: EquivalencySet

```
Set-SCISO [-ISOs] <List<ISO>> -FamilyName <String> -Release <String> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

Changes one or more properties of an ISO object used in a Virtual Machine Manager (VMM) environment. Properties that you can change include:

- Description
- Enabled
- Name
- Owner
- SharePath

### Parameters

#### **-Description<String>**

States a description for the specified object.

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | false |
| Position? | named |

---

|                             |       |
|-----------------------------|-------|
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Enabled<Boolean>**

Enables an object when set to \$True, or disables an object when set to \$False. For example, if you want to upgrade software on a virtual machine template, you can disable the template object in the VMM library to temporarily prevent users from using that object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-FamilyName<String>**

Specifies a family name for a physical resource in the VMM library. This value is used in conjunction with Release, Namespace, and Type to establish equivalency among library resources.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-ISO<ISO>**

Specifies an ISO object.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

|                             |                |
|-----------------------------|----------------|
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-ISOs<List<ISO>>**

Specifies an array of ISO objects.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Name<String>**

Specifies the name of a VMM object.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Owner<String>**

Specifies the owner of a VMM object in the form of a valid domain user account.

Example format: -Owner "Contoso\ReneeLo"

Example format: -Owner "ReneeLo@Contoso"

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-Release<String>**

Specifies a string that describes the release of a library resource. VMM automatically creates a release value for every resource imported into the library. After the resource has been imported, the string can be customized.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-SharePath<String>**

Specifies a path to a valid library share on an existing library server that uses a Universal Naming Convention (UNC) path.

Example format: `-SharePath "\\LibServer01\LibShare"`

|                        |       |
|------------------------|-------|
| Aliases                | none  |
| Required?              | false |
| Position?              | named |
| Default Value          | none  |
| Accept Pipeline Input? | false |

---

|                             |       |
|-----------------------------|-------|
| Accept Wildcard Characters? | false |
|-----------------------------|-------|

### **-UserRole<UserRole>**

Specifies a user role object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

### **Outputs**

The output type is the type of the objects that the cmdlet emits.

- ISO

---

## Notes

- Requires a VMM ISO object, which can be retrieved by using the Get-SCISO cmdlet.

## Examples

### 1: Change the owner of an ISO file.

The first command gets all ISO objects named "OsISO.iso" from LibraryServer01 and stores the ISO objects in an array named \$ISO.

The second command changes the owner of the first ISO object stored in the \$ISO array to Renee Lo, a member of the Contoso.com domain.

```
PS C:\> $ISO = @(Get-SCISO -VMMServer "VMMServer01.Contoso.com" | where { $_.Name -eq "OsISO.iso" -and $_.LibraryServer.Name -eq "LibraryServer01.Contoso.com" } )
PS C:\> Set-SCISO -ISO $ISO[0] -Owner "Contoso\ReneeLo"
```

### 2: Specify an owner for all ISO objects with an "Unknown" owner.

This command gets all ISO objects from the VMM library whose owner is "Unknown", and then specifies an owner for each ISO object.

```
PS C:\> Get-SCISO -VMMServer "VMMServer01.Contoso.com" | where { $_.Owner -eq "Unknown" } | Set-ISO -Owner "Contoso\NevenSokec"
```

### 3: Disable an ISO object stored in the VMM library.

The first command gets the ISO object named OsISO.iso from LibraryServer01 and stores the object in the \$ISO variable.

The second command disables the ISO object stored in the \$ISO variable.

```
PS C:\> $ISO = Get-SCISO -VMMServer "VMMServer01.Contoso.com" | where { $_.Name -eq "OsISO.iso" -and $_.LibraryServer.Name -eq "LibraryServer01.Contoso.com" }
PS C:\> Set-SCISO -ISO $Iso -EnableLibraryObject $FALSE
```

## Related topics

[Get-SCISO](#)

[Remove-SCISO](#)

---

# Set-SCISOConfiguration

---

## Set-SCISOConfiguration

Updates an ISO configuration within a virtual machine configuration.

### Syntax

Parameter Set: Default

```
Set-SCISOConfiguration -ISOConfiguration <ISOConfiguration> [-ISOInstance <ISO> ] [-JobVariable <String> ] [-PinSourceISO <Boolean> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-UseISORemotely <Boolean> ] [ <CommonParameters>]
```

### Detailed Description

The Set-SCISOConfiguration cmdlet updates an ISO configuration within a virtual machine configuration prior to a service deployment.

### Parameters

#### **-ISOConfiguration<ISOConfiguration>**

Specifies an ISO configuration object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

#### **-ISOInstance<ISO>**

Specifies an instance of an ISO object.



|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PinSourceISO<Boolean>**

Indicates whether the source ISO chosen by the user is retained during service deployment configuration.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

---

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-UseISOremotely<Boolean>**

Indicates whether the ISO is stored in a remote location.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **ISOConfiguration**

## Examples

### 1. Set the properties of the ISO configuration for a virtual machine configuration.

The first command gets the service configuration object named Service01 and stores the object in the \$ServiceConfig variable.

The second command gets the computer tier configuration object for the service configuration stored in \$ServiceConfig and stores the object in the \$TierConfig variable.

The third command gets the virtual machine configuration for the computer tier configuration stored in \$TierConfig and stores the object in the \$VMConfig variable.

The fourth command gets the ISO configuration for the virtual machine configuration stored in \$VMConfig and stores the object in the \$ISOConfig variable.

The last command updates the source ISO for the ISO configuration stored in \$ISOConfig and pins the source ISO so that it does not change during service deployment configuration.

```
PS C:\> $ISO = Get-SCISO -Name "TestISO2.iso"
PS C:\> $ServiceConfig = Get-SCServiceConfiguration -Name "Service01"
PS C:\> $TierConfig = Get-SCComputerTierConfiguration -ServiceConfiguration $ServiceConfig
PS C:\> $VMConfig = Get-SCVMConfiguration -ComputerTierConfiguration $TierConfig
PS C:\> $ISOConfig = Get-SCISOConfiguration -VMConfiguration $VMConfig
PS C:\> Set-SCISOConfiguration -ISOConfiguration $ISOConfig -ISOInstance $ISO -PinSourceISO $True
```

## Related topics

[Get-SCISOConfiguration](#)

---

# Set-SCLibraryServer

---

## Set-SCLibraryServer

Sets the properties of a VMM library server.

### Syntax

Parameter Set: Default

```
Set-SCLibraryServer [-LibraryServer] <LibraryServer> [-ClearVMHostGroup] [-Description <String> ] [-EnableUnencryptedFileTransfer <Boolean> ] [-JobGroup <Guid> ] [-JobVariable <String> ] [-LibraryServerManagementCredential <RunAsAccount> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMHostGroup <HostGroup> ] [ <CommonParameters>]
```

Parameter Set: AddVMNetwork

```
Set-SCLibraryServer [-LibraryServer] <LibraryServer> -AddVMNetwork <VMNetwork[]> [-ClearVMHostGroup] [-Description <String> ] [-EnableUnencryptedFileTransfer <Boolean> ] [-JobGroup <Guid> ] [-JobVariable <String> ] [-LibraryServerManagementCredential <RunAsAccount> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMHostGroup <HostGroup> ] [ <CommonParameters>]
```

Parameter Set: RemoveVMNetwork

```
Set-SCLibraryServer [-LibraryServer] <LibraryServer> -RemoveVMNetwork <VMNetwork[]> [-ClearVMHostGroup] [-Description <String> ] [-EnableUnencryptedFileTransfer <Boolean> ] [-JobGroup <Guid> ] [-JobVariable <String> ] [-LibraryServerManagementCredential <RunAsAccount> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMHostGroup <HostGroup> ] [ <CommonParameters>]
```

Parameter Set: VMNetwork

```
Set-SCLibraryServer [-LibraryServer] <LibraryServer> -VMNetworks <VMNetwork[]> [-ClearVMHostGroup] [-Description <String> ] [-EnableUnencryptedFileTransfer <Boolean> ] [-JobGroup <Guid> ] [-JobVariable <String> ] [-LibraryServerManagementCredential <RunAsAccount> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMHostGroup <HostGroup> ] [ <CommonParameters>]
```

### Detailed Description

The Set-SCLibraryServer cmdlet sets the properties of a VMM library server. You can also use this cmdlet as part of a job group, when used with the Add-LibraryShare cmdlet, to add a set of library shares.

---

## Parameters

### **-AddVMNetwork<VMNetwork[]>**

Specifies that one or more VM networks has network connectivity to the specified library server when used in conjunction with Set-SCVMLibraryServer,

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-ClearVMHostGroup**

Resets the host group association for the library server.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Description<String>**

States a description for the specified object.

|                        |       |
|------------------------|-------|
| Aliases                | none  |
| Required?              | false |
| Position?              | named |
| Default Value          | none  |
| Accept Pipeline Input? | false |

|                             |       |
|-----------------------------|-------|
| Accept Wildcard Characters? | false |
|-----------------------------|-------|

### **-EnableUnencryptedFileTransfer<Boolean>**

Indicates, when set to True, that network file transfers do not require encryption. Allowing unencrypted network file transfers can improve performance if neither the source host nor the destination host requires encryption.

Use this parameter to:

- Enable unencrypted file transfers into, or out of, the library.
- Enable unencrypted file transfers into, out of, or within a host group.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobGroup<Guid>**

Specifies an identifier for a series of commands that will run as a set just before the final command that includes the same job group identifier runs.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-LibraryServer<LibraryServer>**

Specifies a VMM library server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-LibraryServerManagementCredential<RunAsAccount>**

Specifies a Run As account for the VMM library.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

---

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RemoveVMNetwork<VMNetwork[]>**

Removes network connectivity for one or more VM networks from the specified library server when used in conjunction with Set-SCVMLibraryServer.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMHostGroup<HostGroup>**

Specifies a virtual machine host group object or an array of host group objects.



|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VMNetworks<VMNetwork[]>**

Specifies that one or more VM networks have network connectivity to the specified library server when used in conjunction with Set-SCVMLibraryServer.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **LibraryServer**

## **Notes**

- Requires a VMM library server object, which can be retrieved by using the Get-SCLibraryServer cmdlet.

---

## Examples

### 1: Change the description of a library server.

The first command gets the library server object named LibraryServer01 on VMMServer01 and stores it in the \$LibServer variable.

The second command changes the description for FileServer01 to "Library server for Production."

```
PS C:\> $LibServer = Get-SCLibraryServer -VMMServer "VMMServer01.Contoso.com" -ComputerName "LibraryServer01.Contoso.com"
```

```
PS C:\> Set-SCLibraryServer -LibraryServer $LibServer -Description "Library server for Production"
```

### 2: Update the description for a library server.

The first command gets the library server object named LibraryServer01 on VMMServer01 and stores it in the \$LibServer variable.

The second command updates the description for the library server object stored in the \$LibServer variable.

```
PS C:\> $LibServer = Get-SCLibraryServer -VMMServer "VMMServer01.Contoso.com" -ComputerName "LibraryServer01.Contoso.com"
```

```
PS C:\> Set-SCLibraryServer -LibraryServer $LibServer -Description "The library server is used by the Seattle office."
```

### 3: Add a VM network to a library server.

The first command gets the VM network object named VMNetwork01 and stores the object in the \$VMNetwork variable.

The second command gets the library server object named LibraryServer01 and uses the pipeline operator to pass the object to the Set-SCLibraryServer which adds VMNetwork01 to the library server.

```
PS C:\> $VMNetwork = Get-SCVMNetwork -Name "VMNetwork01"
```

```
PS C:\> Get-SCLibraryServer -ComputerName "LibraryServer01.Contoso.com" | Set-SCLibraryServer -AddVMNetwork $VMNetwork
```

## Related topics

[Add-SCLibraryServer](#)

[Get-SCLibraryServer](#)

[Remove-SCLibraryServer](#)

---

# Set-SCLibraryShare

---

## Set-SCLibraryShare

Changes the description property of a VMM library share object.

### Syntax

Parameter Set: Default

```
Set-SCLibraryShare [-LibraryShare] <LibraryShare> [-AddDefaultResources] [-Description <String> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

### Detailed Description

The Set-SCLibraryShare cmdlet changes the description property of a VMM library share object.

### Parameters

#### -AddDefaultResources

Indicates that the default resources for a library share are added.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### -Description<String>

States a description for the specified object.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

---

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-LibraryShare<LibraryShare>**

Specifies a VMM library share object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **LibraryShare**

## **Examples**

### **1: Change the description of a library share.**

The first command retrieves the library share object named FileShare01 on LibraryServer01 from the VMM library on VMMServer01 and then stores the library share object in the \$LibShare variable.

The second command changes the description for FileShare01 to "Library share for Test".

```
PS C:\> $LibShare = Get-SCLibraryShare -VMMServer "VMMServer01.Contoso.com" | where {
$_ .LibraryServer.Name -eq "LibraryServer01.Contoso.com" -and $_.Name -eq "FileShare01" }
PS C:\> Set-SCLibraryShare -LibraryShare $LibShare -Description "Library share for Test"
```

---

## **Related topics**

[Add-SCLibraryShare](#)

[Find-SCLibraryShare](#)

[Get-SCLibraryShare](#)

[Read-SCLibraryShare](#)

[Remove-SCLibraryShare](#)

---

# Set-SCLoadBalancer

---

## Set-SCLoadBalancer

Modifies the properties of a load balancer.

### Syntax

Parameter Set: Default

```
Set-SCLoadBalancer [-LoadBalancer] <LoadBalancer> [-AddLogicalNetworkDedicatedIP <LogicalNetwork[]> ] [-AddLogicalNetworkVIP <LogicalNetwork[]> ] [-AddVMHostGroup <HostGroup[]> ] [-ConfigurationProvider <ConfigurationProvider> ] [-JobVariable <String> ] [-LoadBalancerAddress <String> ] [-Manufacturer <String> ] [-Model <String> ] [-Port <UInt16> ] [-PROTipID <Guid> ] [-RemoveLogicalNetworkDedicatedIP <LogicalNetwork[]> ] [-RemoveLogicalNetworkVIP <LogicalNetwork[]> ] [-RemoveVMHostGroup <HostGroup[]> ] [-RunAsAccount <RunAsAccount> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters> ]
```

### Detailed Description

The Set-SCLoadBalancer cmdlet modifies the properties of a load balancer.

### Parameters

#### **-AddLogicalNetworkDedicatedIP<LogicalNetwork[]>**

Specifies the logical network from which an IP address should be assigned to a virtual machine in a service tier as the back-end address for a service.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-AddLogicalNetworkVIP<LogicalNetwork[]>**

Specifies the logical network from which a virtual IP (VIP) address should be assigned to a load balancer VIP as the front-end address for a service.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-AddVMHostGroup<HostGroup[]>**

Adds one or more host groups to an existing host group array or private cloud.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-ConfigurationProvider<ConfigurationProvider>**

Specifies a configuration provider object. A configuration provider is a plug-in to VMM that translates VMM PowerShell commands to API calls that are specific to a type of load balancer. If no configuration provider is specified, VMM uses the Manufacturer and Model information to choose an available configuration provider. If no configuration provider is found, the load balancer will not be added.

|                        |       |
|------------------------|-------|
| Aliases                | none  |
| Required?              | false |
| Position?              | named |
| Default Value          | none  |
| Accept Pipeline Input? | false |



---

|                             |       |
|-----------------------------|-------|
| Accept Wildcard Characters? | false |
|-----------------------------|-------|

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-LoadBalancer<LoadBalancer>**

Specifies a load balancer object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-LoadBalancerAddress<String>**

Specifies the fully qualified domain name (FQDN) or IP address of a load balancer. Usual formats are FQDN, IPv4 or IPv6 addresses, but check with the load balancer manufacturer for the valid format for your load balancer.

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | false |
| Position? | named |

---

|                             |       |
|-----------------------------|-------|
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Manufacturer<String>**

Specifies the name of the company that manufactured a physical device.

Valid characters include: letters (a-z), numbers (0-9), underscore (\_), hyphen(-), dot(.), and single quote(').

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Model<String>**

Specifies the model of a physical device.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Port<UInt16>**

Specifies the network port to use when adding an object or creating a connection. Valid values are: 1 to 4095.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RemoveLogicalNetworkDedicatedIP<LogicalNetwork[]>**

Specifies the logical network from which an IP address was assigned to a virtual machine in a service tier as the front-end address for a service, and should now be removed.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RemoveLogicalNetworkVIP<LogicalNetwork[]>**

Specifies the logical network from which a virtual IP (VIP) address was assigned to a load balancer as the front-end address for a service, and should now be removed.

---

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RemoveVMHostGroup<HostGroup[]>**

Removes one or more host groups from a host group array or private cloud.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsAccount<RunAsAccount>**

Specifies a Run As account that contains credentials with permission to perform this action.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **LoadBalancer**

## **Examples**

### **1: Change the configuration provider for a load balancer.**

The first command gets the load balancer object with the address of LB01.Contoso.com and stores the object in the \$LoadBalancer variable.

The second command gets the configuration provider with the Manufacturer of LB Manufacturer 2 and the model of LB02.

---

The third command updates the configuration provider for the load balancer stored in `$LoadBalancer` to the configuration provider stored in `$NewProvider`.

```
PS C:\> $LoadBalancer = Get-SCLoadBalancer -LoadBalancerAddress "LB01.Contoso.com"
PS C:\> $NewProvider = Get-SCConfigurationProvider | where { $_.Type -eq "LoadBalancer" -and
$_.Manufacturer -eq "LB Manufacturer 2" -and $_.Model -eq "LB02"}
PS C:\> Set-SCLoadBalancer -LoadBalancer $LoadBalancer -ConfigurationProvider $NewProvider -
Manufacturer "LB Manufacturer 2" -Model "LB02"
```

## Related topics

[Add-SCLoadBalancer](#)

[Get-SCLoadBalancer](#)

[Read-SCLoadBalancer](#)

[Remove-SCLoadBalancer](#)

[Test-SCLoadBalancer](#)

---

# Set-SCLoadBalancerConfiguration

---

## Set-SCLoadBalancerConfiguration

Updates a load balancer configuration object for a computer tier.

### Syntax

Parameter Set: Default

```
Set-SCLoadBalancerConfiguration -LoadBalancerConfiguration <LoadBalancerConfiguration> [-JobVariable <String> ] [-LoadBalancer <LoadBalancer> ] [-LoadBalancerVIP <String> ] [-PinLoadBalancer <Boolean> ] [-PinVIPAddressPool <Boolean> ] [-Port <Int32> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-UseExistingVIPAddress <Boolean> ] [-VIPAddressPool <StaticIPAddressPool> ] [ <CommonParameters>]
```

### Detailed Description

The Set-SCLoadBalancerConfiguration cmdlet updates a load balancer configuration object for a computer tier.

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### **-LoadBalancer<LoadBalancer>**

Specifies a load balancer object.

---

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-LoadBalancerConfiguration<LoadBalancerConfiguration>**

Specifies a load balancer configuration object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-LoadBalancerVIP<String>**

Specifies a virtual IP (VIP) in a load balancer.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |



---

## **-PinLoadBalancer<Boolean>**

Indicates whether the load balancer chosen by the user is retained during service deployment configuration.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-PinVIPAddressPool<Boolean>**

Indicates whether the virtual IP (VIP) address pool chosen by the user is retained during service deployment configuration.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-Port<Int32>**

Specifies the network port to use when adding an object or creating a connection. Valid values are: 1 to 4095.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-UseExistingVIPAddress<Boolean>**

Indicates that the existing virtual IP (VIP) address is used, if one has been assigned.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-VIPAddressPool<StaticIPAddressPool>**

Specifies a static IP address pool.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

### **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **LoadBalancerConfiguration**

### **Examples**

#### **1: Set the properties of a load balancer configuration.**

The first command gets the service configuration object named Service01 and stores the object in the \$ServiceConfig variable.

The second command gets the computer tier configuration for the service configuration stored in \$ServiceConfig and stores the object in the \$TierConfig variable.

The third command gets the load balancer configuration for the computer tier configuration stored in \$TierConfig and stores the object in the \$LBConfig variable.

The fourth command gets the load balancer with the address of LB01.Contoso.com and stores the object in the \$LB variable.

The last command set the properties of the load balancer configuration object stored in \$LB.

```
PS C:\> $ServiceConfig = Get-SCServiceConfiguration -Name "Service01"
```

```
PS C:\> $TierConfig = Get-SCComputerTierConfiguration -ServiceConfiguration $ServiceConfig
```

```
PS C:\> $LBConfig = Get-SCLoadBalancerConfiguration -ComputerTierConfiguration $TierConfig
```

```
PS C:\> $LB = Get-SCLoadBalancer -LoadBalancerAddress "LB01.Contoso.com"
```

---

```
PS C:\> Set-SCLoadBalancerConfiguration -LoadBalancerConfiguration $LBConfig -LoadBalancer $LB
```

## Related topics

[Get-SCLoadBalancerConfiguration](#)

---

# Set-SCLoadBalancerTemplate

---

## Set-SCLoadBalancerTemplate

Configures the properties of a load balancer template.

### Syntax

Parameter Set: Default

```
Set-SCLoadBalancerTemplate -LoadBalancerTemplate <LoadBalancerTemplate> [-JobVariable <String> ] [-LoadBalancerVIPTemplate <LoadBalancerVIPTemplate> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMNetworkServiceSetting <String> ] [-VMNetworkVIP <VMNetwork> ] [ <CommonParameters>]
```

Parameter Set: LogicalNetworkVIP

```
Set-SCLoadBalancerTemplate -LoadBalancerTemplate <LoadBalancerTemplate> [-JobVariable <String> ] [-LoadBalancerVIPTemplate <LoadBalancerVIPTemplate> ] [-LogicalNetworkVIP <LogicalNetwork> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMNetworkServiceSetting <String> ] [-VMNetworkVIP <VMNetwork> ] [ <CommonParameters>]
```

### Detailed Description

The Set-SCLoadBalancerTemplate cmdlet configures the properties of a load balancer template.

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-LoadBalancerTemplate<LoadBalancerTemplate>**

Specifies a load balancer template object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-LoadBalancerVIPTemplate<LoadBalancerVIPTemplate>**

Specifies a load balancer virtual IP (VIP) template.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-LogicalNetworkVIP<LogicalNetwork>**

Specifies the logical networks from which the front-end IP address for the load balancer should be assigned (the front-end logical network affinity).

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VMNetworkServiceSetting<String>**

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## -VMNetworkVIP<VMNetwork>

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Examples

### 1: Configure a load balancer template.

The first command gets the service template object named ServiceTemplate01 and stores the object in the \$ServiceTemplate variable.

The second command gets the computer tier template for the service template stored in \$ServiceTemplate and stores the object in the \$TierTemplate variable.

The third command gets the load balancer template for the computer tier stored in \$TierTemplate and stores the object in the \$LoadBalancerTemplate variable.

The last command sets the properties for the load balancer template stored in \$LoadBalancerTemplate.

```
PS C:\> $ServiceTemplate = Get-SCServiceTemplate -Name "ServiceTemplate01"
PS C:\> $TierTemplate = Get-SCComputerTierTemplate -ServiceTemplate $ServiceTemplate
PS C:\> $LoadBalancerTemplate = Get-SCLoadBalancerTemplate -ComputerTierTemplate
$TierTemplate
PS C:\> Set-SCLoadBalancerTemplate -LoadBalancerTemplate $LoadBalancerTemplate
```

## Related topics

[Get-SCComputerTierTemplate](#)

[Get-SCLoadBalancerTemplate](#)

[Get-SCServiceTemplate](#)

[New-SCLoadBalancerTemplate](#)

[Remove-SCLoadBalancerTemplate](#)



---

# Set-SCLoadBalancerVIPTemplate

---

## Set-SCLoadBalancerVIPTemplate

Modifies the properties of a load balancer VIP template.

### Syntax

Parameter Set: Default

```
Set-SCLoadBalancerVIPTemplate [-LoadBalancerVIPTemplate] <LoadBalancerVIPTemplate> [-Description <String> ] [-DisableLoadBalancerConnectionPersistence] [-JobVariable <String> ] [-LoadBalancerConnectionPersistence <LoadBalancerConnectionPersistence> ] [-LoadBalancerHealthMonitor <LoadBalancerHealthMonitor[]> ] [-LoadBalancerManufacturer <String> ] [-LoadBalancerModel <String> ] [-LoadBalancerPort <UInt16> ] [-LoadBalancerProtocol <LoadBalancerProtocol> ] [-LoadBalancingMethod <LoadBalancingMethod> ] [-MakeGeneric] [-Name <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Set-SCLoadBalancerVIPTemplate cmdlet modifies the properties of a load balancer virtual IP (VIP) template.

### Parameters

#### -Description<String>

States a description for the specified object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-DisableLoadBalancerConnectionPersistence**

Indicates whether load balancer connection persistence in a virtual IP (VIP) profile is disabled.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

-

## **LoadBalancerConnectionPersistence<LoadBalancerConnectionPersistence>**

Specifies a load balancer connection persistence object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-LoadBalancerHealthMonitor<LoadBalancerHealthMonitor[]>**

Specifies a load balancer health monitor object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-LoadBalancerManufacturer<String>**

Specifies the name of the company that manufactured a load balancer.

Valid characters include: letters (a-z), numbers (0-9), underscore (\_), hyphen(-), dot(.), and single quote(').

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-LoadBalancerModel<String>**

Specifies the model of a load balancer

|               |       |
|---------------|-------|
| Aliases       | none  |
| Required?     | false |
| Position?     | named |
| Default Value | none  |

|                             |       |
|-----------------------------|-------|
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-LoadBalancerPort<UInt16>**

Specifies the port to use when configuring a virtual IP (VIP) in a load balancer.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-LoadBalancerProtocol<LoadBalancerProtocol>**

Specifies the protocol to use when connecting to a load balancer, or a load balancer protocol object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-LoadBalancerVIPTemplate<LoadBalancerVIPTemplate>**

Specifies a load balancer virtual IP (VIP) template.

|               |      |
|---------------|------|
| Aliases       | none |
| Required?     | true |
| Position?     | 1    |
| Default Value | none |

|                             |                |
|-----------------------------|----------------|
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-LoadBalancingMethod<LoadBalancingMethod>**

Specifies the load balancing method to use. Valid values are: RoundRobin, LeastConnectionsmember, Observedmember, Predictivemember, Ratiomember, Fastestmember, LeastConnections, Observednode, Predictivenode, Rationode, FastestResponseTime, LeastSessions, None

To determine the available methods for a specific load balancer, use the following command: (Get-SCLoadBalancer)[0].AvailableLoadBalancingMethods

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-MakeGeneric**

Indicates that a virtual IP (VIP) profile is able to apply generic load balancer settings.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Name<String>**

Specifies the name of a VMM object.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

---

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

---

|                             |                |
|-----------------------------|----------------|
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **LoadBalancer**

## Examples

### 1: Change the load balancing method in a load balancer virtual IP (VIP) template.

The first command gets the VIP template object named VIPTemplate01 and stores the object in the \$VIPTemplate variable.

The second command creates a new load balancing method object with the name Round Robin and stores the object in the \$LBMethod variable.

The last command changes the load balancing method for the VIP template stored in \$VIPTemplate to the method stored in \$LBMethod, which is Round Robin.

```
PS C:\> $VIPTemplate = Get-SCLoadBalancerVIPTemplate -Manufacturer "LB Manufacturer" -Model "LB01" -Name "VIPTemplate01"
```

```
PS C:\> $LBMethod = New-SCLoadBalancingMethod -Name "RoundRobin"
```

```
PS C:\> Set-SCLoadBalancerVIPTemplate -LoadBalancerVIPTemplate $VIPTemplate -LoadBalancingMethod $LBMethod
```

## Related topics

[Get-SCLoadBalancerVIPTemplate](#)

[New-SCLoadBalancerVIPTemplate](#)

[Remove-SCLoadBalancerVIPTemplate](#)

---

# Set-SCLogicalNetwork

---

## Set-SCLogicalNetwork

Changes the properties of a logical network object

### Syntax

Parameter Set: Default

```
Set-SCLogicalNetwork [-LogicalNetwork] <LogicalNetwork> [-Description <String> ] [-  
EnableNetworkVirtualization <Boolean> ] [-JobVariable <String> ] [-  
LogicalNetworkDefinitionIsolation <Boolean> ] [-Name <String> ] [-PROTipID <Guid> ] [-  
RunAsynchronously] [-UseGRE <Boolean> ] [-VMMServer <ServerConnection> ] [  
<CommonParameters>]
```

### Detailed Description

The Set-SCLogicalNetwork cmdlet changes the properties of a Virtual Machine Manager (VMM) logical network object. Properties that you can update include the name and description of the logical network.

### Parameters

#### **-Description<String>**

States a description for the specified object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### **-EnableNetworkVirtualization<Boolean>**

Specifies whether network virtualization is enabled. The default value is \$False.



|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-LogicalNetwork<LogicalNetwork>**

Specifies a logical network. A logical network is a named grouping of IP subnets and VLANs that is used to organize and simplify network assignments.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

---

## **-LogicalNetworkDefinitionIsolation<Boolean>**

Specifies whether logical network definition isolation is enabled.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-Name<String>**

Specifies the name of a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-UseGRE<Boolean>**

Specifies whether network virtualization Generic Routing Encapsulation (GRE) is used. The default value is \$False.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **LogicalNetwork**

## Notes

- Requires a VMM logical network object, which can be retrieved by using the Get-SCLogicalNetwork cmdlet.

## Examples

### 1: Change the name of a logical network.

The first command gets the logical network object named LogicalNetwork01 and stores the object in the \$LogicalNetwork variable.

The second command changes the name of the logical network stored in \$LogicalNetwork to "LogicalNetwork02".

```
PS C:\> $LogicalNetwork = Get-SCLogicalNetwork -Name "LogicalNetwork01"
```

```
PS C:\> Set-SCLogicalNetwork -LogicalNetwork $LogicalNetwork -Name "LogicalNetwork02"
```

## Related topics

[Get-SCLogicalNetwork](#)

[New-SCLogicalNetwork](#)

[Remove-SCLogicalNetwork](#)

---

# Set-SCLogicalNetworkDefinition

---

## Set-SCLogicalNetworkDefinition

Modifies a logical network definition.

### Syntax

Parameter Set: Default

```
Set-SCLogicalNetworkDefinition [-LogicalNetworkDefinition] <LogicalNetworkDefinition> [-AddVMHostGroup <HostGroup[]> ] [-JobVariable <String> ] [-Name <String> ] [-PROTipID <Guid> ] [-RemoveVMHostGroup <HostGroup[]> ] [-RunAsynchronously] [-SubnetVLAN <SubnetVLAN[]> ] [-VMMServer <ServerConnection> ] [ <CommonParameters> ]
```

### Detailed Description

The Set-SCLogicalNetworkDefinition modifies a logical network definition. For example, you can add a host group to or remove a host group from a logical network definition (also called a network site).

### Parameters

#### **-AddVMHostGroup<HostGroup[]>**

Adds one or more host groups to an existing host group array or private cloud.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-LogicalNetworkDefinition<LogicalNetworkDefinition>**

Specifies a logical network definition (also called a network site) that contains the subnet that the IP address pool serves as specified by the Subnet parameter.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-Name<String>**

Specifies the name of a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

---

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RemoveVMHostGroup<HostGroup[]>**

Removes one or more host groups from a host group array or private cloud.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-SubnetVlan<SubnetVlan[]>**

Specifies one or more IP subnet and VLAN sets.

---

For information about creating a SubnetVLan, type: "Get-Help New-SCSubNetVLan".

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **LogicalNetworkDefiniton**

## **Notes**

- Requires a VMM logical network definition object, which can be retrieved by using the Get-SCLogicalNetworkDefinition cmdlet.



---

## Examples

### 1: Change the host groups associated with a logical network definition.

The first command gets the logical network named "LogicalNetwork01" and stores it in the \$LogicalNetwork variable.

The second command gets the host group named "All Hosts\HostGroup02\Production" and stores it in the \$VMHostGroup variable.

The third command gets the logical network definition named "Logical Network Definition 01" associated with the logical network stored in \$LogicalNetwork and the host group stored in \$VMHostGroup.

The fourth command gets the host group object named "All Hosts\HostGroup-3\Production" and stores the object in the \$HostGroup variable.

The last command adds the host group stored in \$HostGroup to the existing host groups array for the logical network definition stored in \$Definition (Logical Network Definition 01).

```
PS C:\> $LogicalNetwork = Get-SCLogicalNetwork -Name "LogicalNetwork01"
```

```
PS C:\> $VMHostGroup = Get-SCVMHostGroup | where { $_.Path -eq "All Hosts\HostGroup02\Production" }
```

```
PS C:\> $Definition = Get-SCLogicalNetworkDefinition -LogicalNetwork $LogicalNetwork -VMHostGroup $VMHostGroup -Name "Logical Network Definition 01"
```

```
PS C:\> $HostGroup = Get-SCVMHostGroup | where { $_.Path -eq "All Hosts\HostGroup03\Production" }
```

```
PS C:\> Set-SCLogicalNetworkDefinition -LogicalNetworkDefinition $Definition -AddVMHostGroup $HostGroup
```

## Related topics

[Get-SCLogicalNetworkDefinition](#)

[New-SCLogicalNetworkDefinition](#)

[Remove-SCLogicalNetworkDefinition](#)

---

# Set-SCLogicalSwitch

---

## Set-SCLogicalSwitch

Updates the properties of a logical switch.

### Syntax

Parameter Set: WithExtensions

```
Set-SCLogicalSwitch [-LogicalSwitch] <LogicalSwitch> [-Description <String> ] [-EnableSriov <Boolean> ] [-JobVariable <String> ] [-MinimumBandwidthMode {Default | Weight | Absolute | None} ] [-Name <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-SwitchUplinkMode {NoTeam | Team} ] [-VirtualSwitchExtensions <VirtualSwitchExtension[]> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: RemoveAllExtensions

```
Set-SCLogicalSwitch [-LogicalSwitch] <LogicalSwitch> -RemoveAllExtensions [-Description <String> ] [-EnableSriov <Boolean> ] [-JobVariable <String> ] [-MinimumBandwidthMode {Default | Weight | Absolute | None} ] [-Name <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-SwitchUplinkMode {NoTeam | Team} ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Set-SCLogicalSwitch cmdlet updates the properties of a logical switch.

### Parameters

#### **-Description<String>**

States a description for the specified object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

### **-EnableSriov<Boolean>**

Specifies whether single-root I/O virtualization (SR-IOV) is enabled.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-LogicalSwitch<LogicalSwitch>**

Specifies a logical switch object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

---

### **-MinimumBandwidthMode<BandwidthMode>**

Specifies the minimum bandwidth mode for the logical switch. Valid values are: Default, Weight, Absolute, None.

The acceptable values for this parameter are:

|          |  |
|----------|--|
| Default  |  |
| Weight   |  |
| Absolute |  |
| None     |  |

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Name<String>**

Specifies the name of a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

---

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RemoveAllExtensions**

Indicates that all virtual switch extensions are removed from the logical switch.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-SwitchUplinkMode<SwitchUplinkMode>**

The acceptable values for this parameter are:

|        |  |
|--------|--|
| NoTeam |  |
| Team   |  |

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VirtualSwitchExtensions<VirtualSwitchExtension[]>**

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Examples

### 1: Change the minimum bandwidth mode for a logical switch.

This command gets the logical switch object named LogicalSwitch01 and uses the pipeline operator to pass the object to Set-SCLogicalSwitch which changes the minimum bandwidth mode for logical switch object to Weight.

```
PS C:\> Get-SCLogicalSwitch -Name "LogicalSwitch01" | Set-SCLogicalSwitch -  
MinimumBandwidthMode "Weight"
```

### 2: Remove all virtual switch extensions from a logical switch.

This command gets the logical switch object named LogicalSwitch01 and uses the pipeline operator to pass the object to Set-SCLogicalSwitch which removes all virtual switch extensions from the logical switch object.

```
PS C:\> Get-SCLogicalSwitch -Name "LogicalSwitch01" | Set-SCLogicalSwitch -  
RemoveAllExtensions
```

## Related topics

[Get-SCLogicalSwitch](#)

[New-SCLogicalSwitch](#)

[Remove-SCLogicalSwitch](#)

---

# Set-SCMACAddressPool

---

## Set-SCMACAddressPool

Modifies a MAC address pool.

### Syntax

Parameter Set: Default

```
Set-SCMACAddressPool [-MACAddressPool] <MACAddressPool> [-AddVMHostGroup <HostGroup[]> ] [-Description <String> ] [-JobVariable <String> ] [-MACAddressRangeEnd <String> ] [-MACAddressRangeStart <String> ] [-Name <String> ] [-PROTipID <Guid> ] [-RemoveVMHostGroup <HostGroup[]> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Set-SCMACAddressPool cmdlet modifies a Virtual Machine Manager (VMM) MAC address pool. For example, you can add a host group to or remove a host group from a MAC address pool. A MAC address pool can be associated with one or more host groups.

### Parameters

#### **-AddVMHostGroup<HostGroup[]>**

Adds one or more host groups to an existing host group array or private cloud.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### **-Description<String>**

States a description for the specified object.



|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-MACAddressPool<MACAddressPool>**

Specifies a MAC address pool.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-MACAddressRangeEnd<String>**

Specifies the last address in a range of static MAC addresses.

---

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-MACAddressRangeStart<String>**

Specifies the first address in a range of static MAC addresses.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Name<String>**

Specifies the name of a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

---

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RemoveVMHostGroup<HostGroup[]>**

Removes one or more host groups from a host group array or private cloud.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

---

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **MACAddressPool**

## Examples

### 1: Change the host groups associated with a MAC address pool.

The first command gets the MAC address pool object named "MAC Address Pool 01" and stores the object in the \$MACPool variable.

The second command gets the host group object named "All Hosts\HostGroup03\Production" and stores the object the \$HostGroup variable.

The last command updates adds the host group stored in \$HostGroup to the MAC address pool stored in \$MACPool. In this case, MAC Address Pool 01 is now also associated with the host group "All Hosts\HostGroup03\Production" in addition to the host groups it was previously associated with.

```
PS C:\> $MACPool = Get-SCMACAddressPool -Name "MAC Address Pool 01"
```

```
PS C:\> $HostGroup = Get-SCVMHostGroup | where { $_.Path -eq "All  
Hosts\HostGroup03\Production" }
```

```
PS C:\> Set-SCMacAddressPool -MACAddressPool $MACPool -AddVMHostGroup $HostGroup
```

## Related topics

[Get-SCMACAddressPool](#)

[New-SCMACAddressPool](#)

[Remove-SCMACAddressPool](#)

---

# Set-SCNativeUplinkPortProfile

---

## Set-SCNativeUplinkPortProfile

Updates a native uplink port profile.

### Syntax

Parameter Set: Default

```
Set-SCNativeUplinkPortProfile [-NativeUplinkPortProfile] <NativeUplinkPortProfile> [-AddLogicalNetworkDefinition <LogicalNetworkDefinition[]> ] [-Description <String> ] [-EnableNetworkVirtualization <Boolean> ] [-JobVariable <String> ] [-LBFOLoadBalancingAlgorithm {TransportPorts | IPAddresses | MacAddresses | HyperVPort} ] [-LBFOTeamMode {Static | SwitchIndependent | LACP} ] [-Name <String> ] [-PROTipID <Guid> ] [-RemoveLogicalNetworkDefinition <LogicalNetworkDefinition[]> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Set-SCNativeUplinkPortProfile cmdlet updates a native uplink port profile. You can add or remove a logical network definition.

### Parameters

#### **-AddLogicalNetworkDefinition<LogicalNetworkDefinition[]>**

Adds the specified logical network definition object.

To get a logical network definition object, use the Get-SCLogicalNetworkDefinition cmdlet.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

---

## **-Description<String>**

States a description for the specified object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-EnableNetworkVirtualization<Boolean>**

Specifies whether network virtualization is enabled. The default value is \$False.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

-

## **LBFOLoadBalancingAlgorithm<NetworkAdapterLBFOLoadBalancingAlgorithm>**

The acceptable values for this parameter are:

|                |  |
|----------------|--|
| TransportPorts |  |
| IPAddresses    |  |
| MacAddresses   |  |
| HyperVPort     |  |

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-LBFOTeamMode<NetworkAdapterLBFOTeamMode>**

The acceptable values for this parameter are:

|                   |  |
|-------------------|--|
| Static            |  |
| SwitchIndependent |  |
| Lacp              |  |

|                        |       |
|------------------------|-------|
| Aliases                | none  |
| Required?              | false |
| Position?              | named |
| Default Value          | none  |
| Accept Pipeline Input? | false |

---

|                             |       |
|-----------------------------|-------|
| Accept Wildcard Characters? | false |
|-----------------------------|-------|

### **-Name<String>**

Specifies the name of a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-NativeUplinkPortProfile<NativeUplinkPortProfile>**

Specifies a native uplink port profile object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                        |       |
|------------------------|-------|
| Aliases                | none  |
| Required?              | false |
| Position?              | named |
| Default Value          | none  |
| Accept Pipeline Input? | false |



|                             |       |
|-----------------------------|-------|
| Accept Wildcard Characters? | false |
|-----------------------------|-------|

### **-RemoveLogicalNetworkDefinition<LogicalNetworkDefinition[]>**

Removes the specified logical network definition object.

To get a logical network definition object, use the Get-SCLogicalNetworkDefinition cmdlet.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|               |       |
|---------------|-------|
| Aliases       | none  |
| Required?     | false |
| Position?     | named |
| Default Value | none  |

---

|                             |                |
|-----------------------------|----------------|
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Examples

### 1: Remove a logical network definition from a native uplink port profile.

The first command gets the logical network definition object named Logical Network Definition 01 and stores the object in the \$LogNetDefinition variable.

The second command gets the native uplink port profile object named NativeUplinkPortProfile01 and uses the pipeline operator to pass the object to Set-SCNativeUplinkPortProfile which removes Logical Network Definition 01 from NativeUplinkPortProfile01.

```
PS C:\> $LogNetDefinition = Get-SCLogicalNetworkDefinition -Name "Logical Network Definition 01"
```

```
PS C:\> Get-SCNativeUplinkPortProfile -Name "NativeUplinkPortProfile01" | Set-SCNativeUplinkPortProfile -RemoveLogicalNetworkDefinition $LogNetDefinition
```

## Related topics

[Get-SCNativeUplinkPortProfile](#)

[New-SCNativeUplinkPortProfile](#)

[Remove-SCNativeUplinkPortProfile](#)

---

# Set-SCNetworkGateway

---

## Set-SCNetworkGateway

Sets the properties of a network gateway object.

### Syntax

Parameter Set: Default

```
Set-SCNetworkGateway [-NetworkGateway] <NetworkGateway> [-ConfigurationProvider <ConfigurationProvider> ] [-ConnectionString <String> ] [-Description <String> ] [-JobVariable <String> ] [-Manufacturer <String> ] [-Model <String> ] [-Name <String> ] [-PROTipID <Guid> ] [-RunAsAccount <RunAsAccount> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Set-SCNetworkGateway cmdlet updates the properties of a network gateway object.

### Parameters

#### **-ConfigurationProvider<ConfigurationProvider>**

Specifies a configuration provider object. A configuration provider is a plug-in to VMM that translates VMM PowerShell commands to API calls that are specific to a type of load balancer. If no configuration provider is specified, VMM uses the Manufacturer and Model information to choose an available configuration provider. If no configuration provider is found, the load balancer will not be added.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-ConnectionString<String>**

Specifies the information required to connect to the virtual switch extension manager.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-Description<String>**

States a description for the specified object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-Manufacturer<String>**

Specifies the name of the company that manufactured a physical device.

Valid characters include: letters (a-z), numbers (0-9), underscore (\_), hyphen(-), dot(.), and single quote(').

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-Model<String>**

Specifies the model of a physical device.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-Name<String>**

Specifies the name of a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-NetworkGateway<NetworkGateway>**

Specifies a network gateway object.

To obtain a network gateway object, use the Get-SCNetworkGateway cmdlet.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RunAsAccount<RunAsAccount>**

Specifies a Run As account that contains credentials with permission to perform this action.

|                        |       |
|------------------------|-------|
| Aliases                | none  |
| Required?              | false |
| Position?              | named |
| Default Value          | none  |
| Accept Pipeline Input? | false |

|                             |       |
|-----------------------------|-------|
| Accept Wildcard Characters? | false |
|-----------------------------|-------|

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Related topics**

[Add-SCNetworkGateway](#)

[Get-SCNetworkGateway](#)

[Read-SCNetworkGateway](#)

[Remove-SCNetworkGateway](#)

---

[Test-SCNetworkGateway](#)



---

# Set-SCNotification

---

## Set-SCNotification

Dismisses update notifications for a service template or service instance.

### Syntax

Parameter Set: Default

```
Set-SCNotification -Dismiss -Notifications <Notification[]> [-JobVariable <String> ] [-  
PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [  
<CommonParameters>]
```

### Detailed Description

The Set-SCNotification cmdlet dismisses update notifications for a service template or service instance.

### Parameters

#### -Dismiss

Dismisses the error on an object or an update notification on a service instance.

After an error is dismissed, the object is refreshed. If the error reappears, refreshing does not solve the problem and you must fix the error.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### -JobVariable<String>

Specifies that job progress is tracked and stored in the variable named by this parameter.

---

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Notifications<Notification[]>**

Specifies one or more notification objects.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Examples**

### **1: Dismiss all update notifications for a service.**

The first command gets the service object named Service01 and stores the object in the \$Service variable.

The second command gets all notification objects for Service01 and stores the objects in \$Notifications.

The last command dismisses all notifications for Service01.

```
PS C:\> $Service = Get-SCService -Name "Service01"
PS C:\> $Notifications = Get-SCNotification -NotifiedObject $Service
PS C:\> Set-SCNotification -Notifications $Notifications -Dismiss
```

---

## Related topics

[Get-SCNotification](#)

---

# Set-SCOpsMgrConnection

---

## Set-SCOpsMgrConnection

Updates the Operations Manager connection settings.

### Syntax

Parameter Set: Default

```
Set-SCOpsMgrConnection [-EnableMaintenanceModeIntegration <Boolean> ] [-  
EnableOpsMgrConnection <Boolean> ] [-EnablePRO <Boolean> ] [-JobVariable <String> ] [-  
PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [  
<CommonParameters>]
```

### Detailed Description

The Set-SCOpsMgrConnection cmdlet updates the Operations Manager connection. Settings that can be changed include whether the connection is enabled or disabled, and whether maintenance mode integration is enabled for the connection.

### Parameters

#### **-EnableMaintenanceModeIntegration<Boolean>**

Indicates whether maintenance mode integration is enabled for this connection.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### **-EnableOpsMgrConnection<Boolean>**

Indicates whether the connection to the Operations Manager management server is enabled.

---

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-EnablePRO<Boolean>**

Indicates whether PRO is enabled for this connection.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

---

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **OpsMgrConnection**

## Examples

### 1: Disable the Operations Manager connection.

This command disables the Operations Manager connection by setting the EnableOpsMgrConnection to False.

```
PS C:\> Set-SCOpsMgrConnection -EnableOpsMgrConnection $False
```

### 2: Enable the Operations Manager connection.

This command enables the Operations Manager connection by setting the EnableOpsMgrConnection variable to True.

```
PS C:\> Set-SCOpsMgrConnection -EnableOpsMgrConnection $True
```

## Related topics

[Get-SCOpsMgrConnection](#)

[New-SCOpsMgrConnection](#)

[Remove-SCOpsMgrConnection](#)

[Write-SCOpsMgrConnection](#)



---

# Set-SCPpackageMapping

---

## Set-SCPpackageMapping

Updates a package mapping object.

### Syntax

Parameter Set: LocalFile

```
Set-SCPpackageMapping -LocalFile <String> -PackageMapping <PackageMapping> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: PackageFile

```
Set-SCPpackageMapping -PackageMapping <PackageMapping> -UsePackageFileMapping [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: TargetObject

```
Set-SCPpackageMapping -PackageMapping <PackageMapping> [-TargetObject <ClientObject> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Set-SCPpackageMapping cmdlet updates a package mapping object. To create a package mapping object, see New-SCPpackageMapping.

### Parameters

#### **-LocalFile<String>**

Specifies the location of an exported package.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-PackageMapping<PackageMapping>**

Specifies a package mapping object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-TargetObject<ClientObject>**

Specifies the object to which you want to map a resource.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-UsePackageFileMapping**

Indicates that the package file is uploaded.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Examples**

### **1: Update a package mapping object.**

The first command gets the template package at the specified path.

The second command creates a package mapping object for the package stored in \$TemplatePackage and stores the object in the \$Mappings variable.

The third command gets a mapping object by package ID and stores the object in the \$Mapping variable.

The fourth command gets the virtual hard disk object named VHD01 and stores the object in the \$Resource variable.

The last command binds the mapping stored in \$Mapping to the object stored in \$Resource (VHD01).

```
PS C:\> $TemplatePackage = Get-SCTemplatePackage -Path
"C:\TemplateExports\ServiceTemplate01.new.xml"
PS C:\> $Mappings = New-SCPackageMapping -TemplatePackage $TemplatePackage
PS C:\> $Mapping = $Mappings | where {$_.PackageID -eq "VHD01.vhd"}
PS C:\> $Resource = Get-SCVirtualHardDisk -Name "VHD01.vhd"
PS C:\> Set-SCPackageMapping -PackageMapping $Mapping -TargetObject $Resource
```

## **Related topics**

[New-SCPackageMapping](#)

---

# Set-SCPlacementConfiguration

---

## Set-SCPlacementConfiguration

Sets the placement configuration settings for a host group.

### Syntax

Parameter Set: SetToFavor

```
Set-SCPlacementConfiguration -Favor -PlacementConfiguration <PlacementConfigurationSettings> [-DVDDriveRequirement] [-JobVariable <String> ] [-LoadBalancerRequirement] [-NetworkRequirement] [-PassthroughDiskRequirement] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [-VMQueueAvailability] [-Confirm] [-WhatIf] [<CommonParameters>]
```

Parameter Set: SetToInherit

```
Set-SCPlacementConfiguration -Inherit <Boolean> -PlacementConfiguration <PlacementConfigurationSettings> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [-Confirm] [-WhatIf] [<CommonParameters>]
```

Parameter Set: SetToMustMeet

```
Set-SCPlacementConfiguration -MustMeet -PlacementConfiguration <PlacementConfigurationSettings> [-DVDDriveRequirement] [-JobVariable <String> ] [-LoadBalancerRequirement] [-NetworkRequirement] [-PassthroughDiskRequirement] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [-VMQueueAvailability] [-Confirm] [-WhatIf] [<CommonParameters>]
```

Parameter Set: SetToOff

```
Set-SCPlacementConfiguration -Off -PlacementConfiguration <PlacementConfigurationSettings> [-DVDDriveRequirement] [-JobVariable <String> ] [-LoadBalancerRequirement] [-NetworkRequirement] [-PassthroughDiskRequirement] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [-VMQueueAvailability] [-Confirm] [-WhatIf] [<CommonParameters>]
```

Parameter Set: SetToShouldMeet

```
Set-SCPlacementConfiguration -PlacementConfiguration <PlacementConfigurationSettings> -ShouldMeet [-DVDDriveRequirement] [-JobVariable <String> ] [-LoadBalancerRequirement] [-NetworkRequirement] [-PassthroughDiskRequirement] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [-VMQueueAvailability] [-Confirm] [-WhatIf] [<CommonParameters>]
```

---

## Detailed Description

The Set-SCPlacementConfiguration cmdlet sets the placement configuration settings for a host group. To update settings for a host group, that host group must not be inheriting its settings from a parent host group.

## Parameters

### -DVDDriveRequirement

Indicates that the destination host must have the number of physical DVD drives required by a virtual machine for placement. If a specific DVD drive letter has been configured on the virtual machine, the host must have a DVD drive that uses that same drive letter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### -Favor

Indicates that the placement process will select a host even if the host does not meet all requirements; no warning message is displayed to the user.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### -Inherit<Boolean>

Indicates whether settings are inherited from the parent host group.

---

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-LoadBalancerRequirement**

Indicates that the destination host must have access to a load balancer for placement.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-MustMeet**

Indicates that the placement process will not select a host if the host does not meet the requirements.

---

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-NetworkRequirement**

Indicates that the destination host must have virtual switches that connect to each of the logical networks required by a virtual machine for placement.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Off**

Indicates that a placement check is turned off, therefore placement will not consider that metric when determining whether the destination host meets placement metrics.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PassthroughDiskRequirement**

Indicates that a destination host must support passthrough disks for placement.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PlacementConfiguration<PlacementConfigurationSettings>**

Specifies a placement configuration object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.



|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-ShouldMeet**

Indicates that the placement process will select a host even if the host does not meet all requirements; a warning message is displayed to the user.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

---

## **-VMQueueAvailability**

Indicates that a destination host must support network optimizations for placement.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-Confirm**

Prompts you for confirmation before executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **PlacementConfiguration**

## Examples

### 1: Set the placement settings which must be met by a host group.

The first command gets the host group object named HostGroup01 and stores the object in the \$HostGroup variable.

The second command gets the placement configuration object for the host group stored in \$HostGroup and stores the object in the \$PlacementConfig variable.

The last command updates the settings for the placement configuration stored in \$PlacementConfig.

```
PS C:\> $HostGroup = Get-SCVMHostGroup "HostGroup01"
```

```
PS C:\> $PlacementConfig = Get-SCPlacementConfiguration -VMHostGroup $HostGroup
```

```
PS C:\> Set-SCPlacementConfiguration -PlacementConfiguration $PlacementConfig -MustMeet -ClusterReserveRequirement -HighAvailabilityRequirement -IPAddressAvailabilityRequirement
```

### 2: Reset the placement settings for a host group to inherit from the parent host group.

The first command gets the host group object named HostGroup01 and stores the object in the \$HostGroup variable.

The second command gets the placement configuration object for the host group stored in \$HostGroup and stores the object in the \$PlacementConfig variable.

The last command sets the placement configuration stored in \$PlacementConfig to inherit its placement settings from its parent host group.

```
PS C:\> $HostGroup = Get-SCVMHostGroup "HostGroup01"
```

```
PS C:\> $PlacementConfig = Get-SCPlacementConfiguration -VMHostGroup $HostGroup
```

```
PS C:\> Set-SCPlacementConfiguration -PlacementConfiguration $PlacementConfig -Inherit $True
```

### 3: Turn off placement settings for a host group.

The first command gets the host group object named HostGroup01 and stores the object in the \$HostGroup variable.

---

The second command gets the placement configuration object for the host group stored in \$HostGroup and stores the object in the \$PlacementConfig variable.

The last command turns off the specified placement settings for the placement configuration stored in \$PlacementConfig.

```
PS C:\> $HostGroup = Get-SCVMHostGroup "HostGroup01"
```

```
PS C:\> $PlacementConfig = Get-SCPlacementConfiguration -VMHostGroup $HostGroup
```

```
PS C:\> Set-SCPlacementConfiguration -PlacementConfiguration $PlacementConfig -Off -  
ClusterReserveRequirement -HighAvailabilityRequirement -IPAddressAvailabilityRequirement
```

## Related topics

[Get-SCPlacementConfiguration](#)

[Get-SCVMHostGroup](#)

---

# Set-SCPortClassification

---

## Set-SCPortClassification

Sets the properties of a port classification.

### Syntax

Parameter Set: Default

```
Set-SCPortClassification [-PortClassification] <PortClassification> [[-Description] <String> ] [-JobVariable <String> ] [-Name <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Set-SCPortClassification cmdlet updates the properties of a port classification.

### Parameters

#### **-Description<String>**

States a description for the specified object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | 2     |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Name<String>**

Specifies the name of a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PortClassification<PortClassification>**

Specifies the port classification of the switch to which the virtual network adapter connects.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

---

## Examples

### 1: Change the name of a port classification.

This command gets the port classification object named PortClass01 and uses the pipeline operator to pass the object to Set-SCPortClassification which changes the name of the port classification to PortClass02.

```
PS C:\> Get-SCPortClassification -Name "PortClass01" | Set-SCPortClassification -Name "PortClass02"
```

## Related topics

[Get-SCPortClassification](#)

[New-SCPortClassification](#)

[Remove-SCPortClassification](#)



---

# Set-SCPROMonitorConfiguration

---

## Set-SCPROMonitorConfiguration

Updates the properties of a PRO monitor configuration.

### Syntax

Parameter Set: EditSetting

```
Set-SCPROMonitorConfiguration -AutomaticMode <Boolean> -MonitoringEnabled <Boolean> -  
PROMonitorConfiguration <PROMonitorConfiguration> [-JobVariable <String> ] [-PROTipID <Guid>  
] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: Inherit

```
Set-SCPROMonitorConfiguration -Inherit -PROMonitorConfiguration <PROMonitorConfiguration> [-  
JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer  
<ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Set-SCPROMonitorConfiguration cmdlet updates the properties of one or more Performance and Resource Optimization (PRO) monitor configuration objects. Properties that can be set include whether monitoring and automatic remediation are enabled.

### Parameters

#### -AutomaticMode<Boolean>

Indicates that dynamic optimization automatically migrates virtual machines in order to load balance.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-Inherit**

Indicates whether settings are inherited from the parent host group.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-MonitoringEnabled<Boolean>**

Indicates that monitoring is enabled for a PRO monitor.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-PROMonitorConfiguration<PROMonitorConfiguration>**

Specifies a PRO monitor configuration object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

### **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **PROMonitorConfiguration**

### **Examples**

#### **1: Enable monitoring and automatic remediation for a specific PRO monitor.**

The first command gets the PRO monitor object with the specified name and management pack name and stores the object in the \$PROMonitor variable.

The second command gets the host object named VMHost01 and stores the object in the \$VMHost variable.

The third command gets the PRO monitor configuration object for the PRO monitor stored in \$PROMonitor on VMHost01 and stores the object in the \$PROMonitorConfig variable.

The last command enables monitoring and automatic remediation for the PRO monitor configuration stored in \$PROMonitorConfig.

```
PS C:\> $PROMonitor = Get-SCPROMonitor -Name "System Center Virtual Machine Manager Maximum Dynamic Memory Monitor" -ManagementPackName "System Center Virtual Machine Manager PRO V2 HyperV Host Performance"
```

```
PS C:\> $VMHost = Get-SCVMHost -ComputerName "VMHost01.Contoso.com"
```

```
PS C:\> $PROMonitorConfig = Get-SCPROMonitorConfiguration -PROMonitor $PROMonitor -VMHost $VMHost
```

---

```
PS C:\> Set-SCPROMonitorConfiguration -PROMonitorConfiguration $PROMonitorConfig -
MonitoringEnabled $True -AutomaticMode $True
```

## 2: Disable automatic remediation for all PRO monitors on a specified host.

The first command gets the host object named VMHost01 and stores the object in the \$VMHost variable.

The second command gets all PRO monitor configuration objects on VMHost01 and stores the objects in the \$PROMonitorConfigs object array.

The last command uses the ForEach statement to iterate through each PRO monitor configuration object stored in \$PROMonitorConfigs and disables automatic remediation for each monitor configuration.

```
PS C:\> $VMHost = Get-SCVMHost -ComputerName "VMHost01.Contoso.com"
```

```
PS C:\> $PROMonitorConfigs = @(Get-SCPROMonitorConfiguration -VMHost $VMHost)
```

```
PS C:\> ForEach ($PROMonitorConfig in $PROMonitorConfigs) {Set-SCPROMonitorConfiguration -
PROMonitorConfiguration $PROMonitorConfig -MonitoringEnabled $True -AutomaticMode $False}
```

## Related topics

[Get-SCPROMonitor](#)

[Get-SCPROMonitorConfiguration](#)

---

# Set-SCPROTip

---

## Set-SCPROTip

Sets the status of a PRO tip.

### Syntax

Parameter Set: Default

```
Set-SCPROTip -PROTipID <String> [-ActionDetails <String> ] [-ActionDetailsOpsMgrString <String[]> ] [-ActionScript <String> ] [-ActionSummary <String> ] [-ActionSummaryOpsMgrString <String[]> ] [-JobVariable <String> ] [-LastError <String> ] [-LastErrorOpsMgrString <String[]> ] [-RunAsynchronously] [-TipStatus <String> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Set-SCPROTip cmdlet sets the status of a Performance and Resource Optimization (PRO) tip object. This cmdlet, which is called by PRO tip implementation actions and is for use in building PRO Packs, is used by Virtual Machine Manager (VMM) to update the status of a PRO tip while performing the action recommended by the PRO tip. You can use this cmdlet to manually update the status of PRO tips.

### Parameters

#### **-ActionDetails<String>**

Provides a detailed description of what implementing this PRO tip will do.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-ActionDetailsOpsMgrString<String[]>**

Specifies an array of strings used to provide translated action details text. The first element of the array should be the GUID of the Operations Manager string and the following elements should be the parameters for string formatting.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-ActionScript<String>**

Specifies the script that will run by implementing this PRO tip.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-ActionSummary<String>**

Provides a summary description of what implementing this PRO tip will do.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-ActionSummaryOpsMgrString<String[]>**

Specifies an array of strings used to provide translated action summary text. The first element of the array should be the GUID of the Operations Manager string and the following elements should be the parameters for string formatting.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-LastError<String>**

Specifies the error text of a runtime error from a PRO tip script.

|                        |       |
|------------------------|-------|
| Aliases                | none  |
| Required?              | false |
| Position?              | named |
| Default Value          | none  |
| Accept Pipeline Input? | false |



|                             |       |
|-----------------------------|-------|
| Accept Wildcard Characters? | false |
|-----------------------------|-------|

### **-LastErrorOpsMgrString<String[]>**

Specifies an array of strings used to provide translated error text. The first element of the array should be the GUID of the Operations Manager string and the following elements should be the parameters for string formatting.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<String>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | false |
| Position? | named |

|                             |       |
|-----------------------------|-------|
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-TipStatus<String>**

Specifies the current status of a PRO tip object.

#### VALID VALUE DESCRIPTION

-----

Active The user can invoke the tip's recommended action.

Initialized The tip has been invoked; any incomplete jobs are queued.

Auto

Running The tip has been invoked; its jobs are running

Resolved The implementation of the tip has completed successfully.

Failed The implementation of the tip has failed.

Dismissed The user has chosen to ignore the tip.

Closed

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                        |                |
|------------------------|----------------|
| Aliases                | none           |
| Required?              | false          |
| Position?              | named          |
| Default Value          | none           |
| Accept Pipeline Input? | True (ByValue) |

---

|                             |
|-----------------------------|
| Accept Wildcard Characters? |
|-----------------------------|

|       |
|-------|
| false |
|-------|

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **PROTip**

## Examples

### 1: Set the status of a PRO tip.

The first command gets all active PRO tip objects from the VMM database and stores the objects in the \$AllPROTips object array.

The last command updates the first tip stored in \$PROTips (as designated by the [0]) to the status "Running".

```
PS C:\> $PROTips = Get-SCPROTip
```

```
PS C:\> Set-SCPROTip -PROTipID $PROTips[0].Id -TipStatus Running
```

## Related topics

[Clear-SCPROTip](#)

[Get-SCPROTip](#)

[Invoke-SCPROTip](#)

[Test-SCPROTip](#)

---

# Set-SCRunAsAccount

---

## Set-SCRunAsAccount

Modifies the properties of a Run As account.

### Syntax

Parameter Set: Default

```
Set-SCRunAsAccount [-RunAsAccount] <RunAsAccount> [-Credential <PSCredential> ] [-Description <String> ] [-JobGroup <Guid> ] [-JobVariable <String> ] [-Name <String> ] [-NoValidation] [-Owner <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-UserRole <UserRole> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Set-SCRunAsAccount cmdlet modifies the properties of a Virtual Machine Manager (VMM) Run As account.

### Parameters

#### **-Credential<PSCredential>**

Specifies a credential object or, for some cmdlets, a Run As account object that contains the user name and password of an account that has permission to perform this action. Or, in the case of Restart-SCJob, has permission to complete a restarted task.

For more information about the PSCredential object, type: "Get-Help Get-Credential".

For more information about Run As accounts, type: "Get-Help New-SCRunAsAccount".

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

---

## **-Description<String>**

States a description for the specified object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-JobGroup<Guid>**

Specifies an identifier for a series of commands that will run as a set just before the final command that includes the same job group identifier runs.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-Name<String>**

Specifies the name of a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-NoValidation**

Indicates that the Run As account will not validate the provided domain credentials.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-Owner<String>**

Specifies the owner of a VMM object in the form of a valid domain user account.

Example format: -Owner "Contoso\ReneeLo"

Example format: -Owner "ReneeLo@Contoso"

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RunAsAccount<RunAsAccount>**

Specifies a Run As account that contains credentials with permission to perform this action.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-UserRole<UserRole>**

Specifies a user role object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **RunAsAccount**



---

## Examples

### 1: Change the name of an existing Run As account.

The first command gets the Run As account object named RunAsAccount01 and stores the object in the \$RunAsAccount variable.

The second command changes the name of the Run As account stored in \$RunAsAccount to "New Run As Account Name" and adds a description for the account.

```
PS C:\> $RunAsAccount = Get-SCRunAsAccount -Name "RunAsAccount01"
```

```
PS C:\> Set-SCRunAsAccount -RunAsAccount $RunAsAccount -Name "New Run As Account Name" -  
Description "This is an administrator account for accessing Hyper-V hosts."
```

## Related topics

[Get-SCRunAsAccount](#)

[New-SCRunAsAccount](#)

---

# Set-SCScript

---

## Set-SCScript

Changes the properties of a script stored in the VMM library.

### Syntax

Parameter Set: Default

```
Set-SCScript [-Script] <Script> [-Description <String> ] [-Enabled <Boolean> ] [-FamilyName <String> ] [-JobVariable <String> ] [-Name <String> ] [-Owner <String> ] [-PROTipID <Guid> ] [-Release <String> ] [-RunAsynchronously] [-SharePath <String> ] [-UserRole <UserRole> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: EquivalencySet

```
Set-SCScript [-Scripts] <List<Script>> -FamilyName <String> -Release <String> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Set-SCScript cmdlet changes one or more properties of a script stored in the VMM library.

Properties that you can change include:

- Description
- Enabled
- Name
- Owner
- SharePath

Script objects represent script files stored in a library share on a library server. Typically, these scripts are either Windows PowerShell scripts or answer files (such as a Sysprep.inf or an Unattend.xml file) that contain the inputs required for the Windows Setup program.

### Parameters

#### **-Description<String>**

States a description for the specified object.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Enabled<Boolean>**

Enables an object when set to \$True, or disables an object when set to \$False. For example, if you want to upgrade software on a virtual machine template, you can disable the template object in the VMM library to temporarily prevent users from using that object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-FamilyName<String>**

Specifies a family name for a physical resource in the VMM library. This value is used in conjunction with Release, Namespace, and Type to establish equivalency among library resources.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Name<String>**

Specifies the name of a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Owner<String>**

Specifies the owner of a VMM object in the form of a valid domain user account.

Example format: -Owner "Contoso\ReneeLo"

Example format: -Owner "ReneeLo@Contoso"

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-Release<String>**

Specifies a string that describes the release of a library resource. VMM automatically creates a release value for every resource imported into the library. After the resource has been imported, the string can be customized.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-Script<Script>**

Specifies a VMM script object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-Scripts<List<Script>>**

Specifies an array of script objects.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-SharePath<String>**

Specifies a path to a valid library share on an existing library server that uses a Universal Naming Convention (UNC) path.

Example format: `-SharePath "\\LibServer01\LibShare"`

|               |       |
|---------------|-------|
| Aliases       | none  |
| Required?     | false |
| Position?     | named |
| Default Value | none  |

|                             |       |
|-----------------------------|-------|
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-UserRole<UserRole>**

Specifies a user role object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

### **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **Script**

---

## Notes

- Requires a VMM script object, which can be retrieved by using the Get-SCScript cmdlet.

## Examples

### 1: Change the description of a script.

The first command retrieves the answer file script object named Sysprep.inf from the library on VMMServer01 and then stores the object in the \$Script variable.

The second command changes the description of this script object to "Windows Server 2008 R2 Sysprep Answer File".

```
PS C:\> $Script = Get-SCScript -VMMServer "VMMServer01.Contoso.com" | where {
$.LibraryServer.Name -eq "LibraryServer01.Contoso.com" -and $_.Name -eq "Sysprep.inf" }
PS C:\> Set-SCScript -Script $Script -Description "Windows Server 2008 R2 Sysprep Answer
File"
```

### 2: Disable a Windows PowerShell script stored in the VMM library.

The first command gets the the PowerShell script object named AddHost.ps1 and stores the object in the \$Script variable.

The second command disables the script stored in \$Script.

```
PS C:\> $Script = Get-SCScript -VMMServer "VMMServer01.Contoso.com" | where {
$.LibraryServer.Name -eq "LibraryServer01.Contoso.com" -and $_.Name -eq "AddHost.ps1" }
PS C:\> Set-SCScript -Script $Script -Enabled $FALSE
```

### 3: Specify an owner for all scripts with an "Unknown" owner

This command gets all script objects from the VMM library whose owner is "Unknown", and then specifies an owner for each script object.

```
PS C:\> Get-SCScript -VMMServer "VMMServer01.Contoso.com" | where {$_.Owner -eq "Unknown"} |
Set-SCScript -Owner "Contoso\ReneeLo"
```

## Related topics

[Get-SCScript](#)

[Remove-SCScript](#)



---

# Set-SCScriptCommand

---

## Set-SCScriptCommand

Configures a script command.

### Syntax

Parameter Set: Default

```
Set-SCScriptCommand [-ScriptCommand] <SCScriptCommand> [-CommandParameters <String> ] [-DeploymentOrder <Int32> ] [-Executable <String> ] [-JobVariable <String> ] [-LibraryResource <CustomResource> ] [-PROTipID <Guid> ] [-RunAsAccount <VMMCredential> ] [-RunAsynchronously] [-ScriptCommandSetting <SCScriptCommandSetting> ] [-ScriptType {PreService | PostService | SaveState | RestoreState | PreInstall | PostInstall | PreUninstall | PostUninstall | OrderedCommand} ] [-StandardInput <String> ] [-TimeoutSeconds <Int32> ] [ <CommonParameters> ]
```

### Detailed Description

The Set-SCScriptCommand cmdlet configures a script command.

### Parameters

#### **-CommandParameters<String>**

Specifies the parameters for a script or executable program.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-DeploymentOrder<Int32>**

Specifies the order in which a computer tier, application host, or application is deployed.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-Executable<String>**

Specifies the name of an executable program.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-LibraryResource<CustomResource>**

Specifies a resource stored in the VMM library.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RunAsAccount<VMMSCredential>**

Specifies a Run As account that contains credentials with permission to perform this action.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-ScriptCommand<SCScriptCommand>**

Specifies a script command object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-ScriptCommandSetting<SCScriptCommandSetting>**

Specifies a script command setting object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-ScriptType<ScriptCommandType>**

Specifies a script type. Valid values are: PreInstall, PostInstall, SaveState, RestoreState, PreService, PostService, PreUninstall, PostUninstall.

The acceptable values for this parameter are:

|                |  |
|----------------|--|
| PreService     |  |
| PostService    |  |
| SaveState      |  |
| RestoreState   |  |
| PreInstall     |  |
| PostInstall    |  |
| PreUninstall   |  |
| PostUninstall  |  |
| OrderedCommand |  |

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-StandardInput<String>**

Specifies a path to a file that contains standard input information to use with the script command.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-TimeoutSeconds<Int32>**

Specifies the amount of time, in seconds, that a process waits before timing out.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **ScriptCommand**

## **Examples**

### **1: Add a custom resource to a script command.**

The first command gets the application profile object named SvcWebAppProfile01 and stores the object in the \$AppProfile variable.

The second command gets the script command object named PreInstall and stores the object in the \$ScriptCommand variable.

The third command gets the resource object named CustomResource and stores the object in the \$Resource variable.

The last command adds the resource object stored in \$Resource to the script command object stored in \$ScriptCommand.

```
PS C:\> $AppProfile = Get-SCApplicationProfile -Name "SvcWebAppProfile01"
PS C:\> $ScriptCommand = Get-SCScriptCommand -ApplicationProfile $AppProfile | Where
{$_.Name -eq "PreInstall"}
PS C:\> $Resource = Get-SCCustomResource -Name "CustomResource.cr"
PS C:\> Set-SCScriptCommand -ScriptCommand $ScriptCommand -LibraryResource $Resource
```

---

## Related topics

[Add-SCScriptCommand](#)

[Get-SCScriptCommand](#)

[Remove-SCScriptCommand](#)

---

# Set-SCScriptCommandSetting

---

## Set-SCScriptCommandSetting

Configures a script command setting.

### Syntax

Parameter Set: DefaultParamSet

```
Set-SCScriptCommandSetting [-ScriptCommandSetting] <SCScriptCommandSetting> [-AlwaysReboot <Boolean> ] [-CommandMayReboot] [-FailOnMatch] [-MatchExitCode <String> ] [-MatchRebootExitCode <String> ] [-MatchStandardError <String> ] [-MatchStandardOutput <String> ] [-PersistStandardErrorPath <String> ] [-PersistStandardOutputPath <String> ] [-RestartOnRetry <Boolean> ] [-RestartScriptOnExitCodeReboot <Boolean> ] [-WarnAndContinueOnMatch] [-WorkingDirectory <String> ] [ <CommonParameters>]
```

### Detailed Description

The Set-SCScriptCommandSetting cmdlet configures a script command setting.

### Parameters

#### **-AlwaysReboot<Boolean>**

Indicates whether a computer or virtual machine should always restart after the script has finished running.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |



---

## **-CommandMayReboot**

Indicates that the script command may reboot the computer or virtual machine.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-FailOnMatch**

Indicates that the action taken when a failure policy is matched is to fail.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-MatchExitCode<String>**

Specifies the failure policy exit code.

Example format: `-MatchExitCode "[1-9][0-9]*"`

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-MatchRebootExitCode<String>**

Specifies the restart policy match exit code.

Example format: `-MatchRebootExitCode "{1641}{{3010}}{{3011}}"`

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-MatchStandardError<String>**

Specifies the failure policy standard error.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-MatchStandardOutput<String>**

Specifies the failure policy standard output.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-PersistStandardErrorPath<String>**

Specifies the file path to store the standard error.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-PersistStandardOutputPath<String>**

Specifies the file path to store the standard output.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RestartOnRetry<Boolean>**

Specifies that a script is restarted upon VMM job restart if the previous job failure was due to a script failure when used in conjunction with New-ScriptCommandSetting or Set-SCScriptCommandSetting.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-RestartScriptOnExitCodeReboot<Boolean>**

Indicates whether the script restarts after the computer or virtual machine is restarted when an exit code is matched.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-ScriptCommandSetting<SCScriptCommandSetting>**

Specifies a script command setting object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-WarnAndContinueOnMatch**

Indicates that the action taken when a failure policy is matched is to warn the user and continue the operation.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-WorkingDirectory<String>**

Specifies a working directory for a script command.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **ScriptCommandSetting**

## **Examples**

### **1: Update the working directory associated with script command.**

The first command gets the application profile object named SvcWebAppProfile01 and stores the object in the \$AppProfile variable.

The second command gets the script command named PreInstall for the application profile stored in \$AppProfile, and then stores the object in the \$ScriptCommand variable.

The third command gets the script command setting object for the script command stored in \$ScriptCommand and stores the object in the \$ScriptCmdSetting variable.

The fourth command sets the working directory setting to Working\_Folder\_03.

The last command updates the script command stored in \$ScriptCommand with the settings stored in \$ScriptCmdSetting.

```
PS C:\> $AppProfile = Get-SCApplicationProfile -Name "SvcWebAppProfile01"
PS C:\> $ScriptCommand = Get-SCScriptCommand -ApplicationProfile $AppProfile | where
{$_ .Name -eq "PreInstall"}
PS C:\> $ScriptCmdSetting = Get-SCScriptCommandSetting -ScriptCommand $ScriptCommand
```

---

```
PS C:\> Set-SCScriptCommandSetting -ScriptCommandSetting $ScriptCmdSetting -WorkingDirectory  
"Working_Folder_03"  
PS C:\> Set-SCScriptCommand -ScriptCommand $ScriptCommand -ScriptCommandSetting  
$ScriptCmdSetting
```

## Related topics

[Get-SCScriptCommandSetting](#)

[New-SCScriptCommandSetting](#)

---

# Set-SCService

---

## Set-SCService

Modifies a VMM service instance.

### Syntax

Parameter Set: UnspecifiedSet

```
Set-SCService [-Service] <Service> [-CostCenter <String> ] [-Description <String> ] [-JobVariable <String> ] [-Name <String> ] [-Owner <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-ServicePriority {Normal | Low | High} ] [-UserRole <UserRole> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: CommitST

```
Set-SCService [-Service] <Service> -CommitPendingServiceTemplate [-CostCenter <String> ] [-Description <String> ] [-JobVariable <String> ] [-Name <String> ] [-Owner <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-ServicePriority {Normal | Low | High} ] [-UserRole <UserRole> ] [-VMMServer <ServerConnection> ] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: DismissST

```
Set-SCService [-Service] <Service> -DismissPendingServiceTemplate [-CostCenter <String> ] [-Description <String> ] [-JobVariable <String> ] [-Name <String> ] [-Owner <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-ServicePriority {Normal | Low | High} ] [-UserRole <UserRole> ] [-VMMServer <ServerConnection> ] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: PendingST

```
Set-SCService [-Service] <Service> -PendingServiceTemplate <ServiceTemplate> [-CostCenter <String> ] [-Description <String> ] [-JobVariable <String> ] [-Name <String> ] [-Owner <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-ServicePriority {Normal | Low | High} ] [-UserRole <UserRole> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

## Detailed Description

The Set-SCService cmdlet modifies a Virtual Machine Manager (VMM) service instance.

## Parameters

### -CommitPendingServiceTemplate

Applies the pending service template to the service instance.

---

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-CostCenter<String>**

Specifies the cost center for a virtual machine so that you can collect data about the allocation of virtual machines (or resources allocated to virtual machines) to make use of in your billing system.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Description<String>**

States a description for the specified object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-DismissPendingServiceTemplate**

Removes a pending service template from a service instance.



---

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Name<String>**

Specifies the name of a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Owner<String>**

Specifies the owner of a VMM object in the form of a valid domain user account.

---

Example format: -Owner "Contoso\ReneeLo"

Example format: -Owner "ReneeLo@Contoso"

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PendingServiceTemplate<ServiceTemplate>**

Specifies a service template object that has been updated but not applied to the service instance.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-Service<Service>**

Specifies a VMM service object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-ServicePriority<ServicePriority>**

Specifies the priority for a service. Valid values are: Normal, Low, High. Default value: Normal.

The acceptable values for this parameter are:

|        |  |
|--------|--|
| Normal |  |
| Low    |  |
| High   |  |

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | false |

|                             |       |
|-----------------------------|-------|
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-UserRole<UserRole>**

Specifies a user role object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | false |

---

|                             |       |
|-----------------------------|-------|
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **Service**

## Examples

### 1: Change the description and priority of a service.

The first command gets the service object named Service01 and stores the object in the \$Service variable.

The second command updates the description and priority for the service stored in \$Service.

```
PS C:\> $Service = Get-SCService -Name "Service01"
```

```
PS C:\> Set-SCService -Service $Service -Description "Contoso Custom Service" -  
ServicePriority Normal
```

### 2. Dismiss a pending service template

The first command gets the service object named Service01 and stores the object in the \$Service variable.

The second command dismisses the pending servicing operation

```
PS C:\> $Service = Get-SCService -Name "Service01"
```

```
PS C:\> Set-SCService -Service $Service -DismissPendingServiceTemplate
```

## Related topics

[Get-SCService](#)

[New-SCService](#)

[Read-SCService](#)

---

[Remove-SCService](#)

[Resume-SCService](#)

[Start-SCService](#)

[Stop-SCService](#)

[Suspend-SCService](#)

[Update-SCService](#)

---

# Set-SCServiceConfiguration

---

## Set-SCServiceConfiguration

Modifies the properties of an undeployed service configuration object stored in the VMM library.

### Syntax

Parameter Set: HostGroup

```
Set-SCServiceConfiguration -ServiceConfiguration <ServiceConfiguration> [-CostCenter <String> ] [-Description <String> ] [-JobVariable <String> ] [-Name <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-ServicePriority <String> ] [-Tag <String> ] [-VMHostGroup <HostGroup> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: Cloud

```
Set-SCServiceConfiguration -Cloud <Cloud> -ServiceConfiguration <ServiceConfiguration> [-CostCenter <String> ] [-Description <String> ] [-JobVariable <String> ] [-Name <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-ServicePriority <String> ] [-Tag <String> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Set-SCServiceConfiguration cmdlet modifies the properties of an undeployed service configuration object stored in the VMM library.

### Parameters

#### **-Cloud<Cloud>**

Specifies a private cloud object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

### **-CostCenter<String>**

Specifies the cost center for a virtual machine so that you can collect data about the allocation of virtual machines (or resources allocated to virtual machines) to make use of in your billing system.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Description<String>**

States a description for the specified object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |



---

## **-Name<String>**

Specifies the name of a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-ServiceConfiguration<ServiceConfiguration>**

Specifies a service configuration object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-ServicePriority<String>**

Specifies the priority for a service. Valid values are: Normal, Low, High. Default value: Normal.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-Tag<String>**

Specifies a word or phrase to associate with an object so that you can search for all objects with the specified set of tags. You can search for a subset of tags, or you can search for the full set of tags.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-VMHostGroup<HostGroup>**

Specifies a virtual machine host group object or an array of host group objects.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **ServiceConfiguration**

---

## Examples

### 1: Set the priority for a Service Configuration object in a cloud.

The first command gets the service configuration object named Service02 and stores the object in the \$SvcConfig variable.

The second command gets the private cloud object named Production and stores the object in the \$Cloud variable.

The last command sets the priority for the service configuration object stored in \$SvcConfig to Normal.

```
PS C:\> $SvcConfig = Get-SCServiceConfiguration -Name "Service02"
```

```
PS C:\> $Cloud = Get-SCCloud -Name "Production"
```

```
PS C:\> Set-SCServiceConfiguration -ServiceConfiguration $SvcConfig -Cloud $Cloud -  
ServicePriority Normal
```

## Related topics

[Get-SCServiceConfiguration](#)

[New-SCServiceConfiguration](#)

[Remove-SCServiceConfiguration](#)

[Update-SCServiceConfiguration](#)

---

# Set-SCServiceSetting

---

## Set-SCServiceSetting

Modifies a service setting.

### Syntax

Parameter Set: Value

```
Set-SCServiceSetting [-ServiceSetting] <ServiceSetting> [-Description <String> ] [-IsEncrypted <Boolean> ] [-IsRequired <Boolean> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-Value <String> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: SecureValue

```
Set-SCServiceSetting [-ServiceSetting] <ServiceSetting> [-Description <String> ] [-IsEncrypted <Boolean> ] [-IsRequired <Boolean> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-SecureValue <SecureString> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Set-SCServiceSetting cmdlet modifies a service setting.

### Parameters

#### **-Description<String>**

States a description for the specified object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

### **-IsEncrypted<Boolean>**

Indicates that a service setting is encrypted.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-IsRequired<Boolean>**

Indicates that a service setting is mandatory.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-SecureValue<SecureString>**

Specifies the value for a secure string.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-ServiceSetting<ServiceSetting>**

Specifies a service setting object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-Value<String>**

Specifies a string used to attribute an object or property.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |



---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Examples

### 1. Make a service setting mandatory.

The first command gets the service template object named ServiceTemplate01 with a release of Beta and stores the object in the \$ServiceTemplate variable.

The second command gets the service setting object named Setting01 from ServiceTemplate01 and stores the object in the \$ServiceSetting variable.

The last command modifies the service setting so that it is mandatory.

```
PS C:\> $Template = Get-SCServiceTemplate -Name "ServiceTemplate01" | where {$_.Release -eq "Beta"}
```

```
PS C:\> $ServiceSetting = Get-SCServiceSetting -ServiceTemplate $Template -Name "Setting01"
```

```
PS C:\> Set-SCServiceSetting -ServiceSetting $ServiceSetting -IsRequired $True
```

## Related topics

[Get-SCServiceSetting](#)

---

# Set-SCServiceTemplate

---

## Set-SCServiceTemplate

Configures the properties of a service template.

### Syntax

Parameter Set: Default

```
Set-SCServiceTemplate [-ServiceTemplate] <ServiceTemplate> [-Description <String> ] [-  
JobVariable <String> ] [-Name <String> ] [-Owner <String> ] [-PROTipID <Guid> ] [-Published  
<Boolean> ] [-Release <String> ] [-RunAsynchronously] [-ServicePriority {Normal | Low |  
High} ] [-UseAsDefaultRelease <Boolean> ] [-UserRole <UserRole> ] [ <CommonParameters>]
```

### Detailed Description

The Set-SCServiceTemplate cmdlet configures the properties of a service template.

For more information about service templates, type: "Get-Help New-SCServiceTemplate -detailed".

### Parameters

#### **-Description<String>**

States a description for the specified object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Name<String>**

Specifies the name of a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Owner<String>**

Specifies the owner of a VMM object in the form of a valid domain user account.

Example format: -Owner "Contoso\ReneeLo"

Example format: -Owner "ReneeLo@Contoso"

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-Published<Boolean>**

Indicates whether a service template should be published.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-Release<String>**

Specifies a string that describes the release of a library resource. VMM automatically creates a release value for every resource imported into the library. After the resource has been imported, the string can be customized.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-ServicePriority<ServicePriority>**

Specifies the priority for a service. Valid values are: Normal, Low, High. Default value: Normal.

The acceptable values for this parameter are:

|        |  |
|--------|--|
| Normal |  |
| Low    |  |
| High   |  |

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-ServiceTemplate<ServiceTemplate>**

Specifies a service template object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-UseAsDefaultRelease<Boolean>**

Specifies that this release is used as the default release for the service template.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-UserRole<UserRole>**

Specifies a user role object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

---

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **ServiceTemplate**

## Examples

### 1: Set the priority for a service template.

The first command gets the service template object named ServiceTemplate01 and stores the object in the \$SvcTemplate variable.

The second command sets the priority for the service template object stored in \$SvcTemplate to High.

```
PS C:\> $SvcTemplate = Get-SCServiceTemplate -Name "ServiceTemplate01"
```

```
PS C:\> Set-SCServiceTemplate -ServiceTemplate $SvcTemplate -ServicePriority High
```

## Related topics

[Get-SCServiceTemplate](#)

[New-SCServiceTemplate](#)

[Read-SCServiceTemplate](#)

[Remove-SCServiceTemplate](#)

[Resolve-SCServiceTemplate](#)

[Test-SCServiceTemplate](#)

---

# Set-SCServicingWindow

---

## Set-SCServicingWindow

Modifies the properties of a servicing window.

### Syntax

Parameter Set: NoFrequencyUpdate

```
Set-SCServicingWindow [-ServicingWindow] <ServicingWindow> [-Category <String> ] [-Description <String> ] [-JobVariable <String> ] [-MinutesDuration <Int32> ] [-Name <String> ] [-Owner <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-StartDate <DateTime> ] [-StartTimeOfDay <DateTime> ] [-TimeZone <Int32> ] [ <CommonParameters>]
```

Parameter Set: DailyFrequency

```
Set-SCServicingWindow [-ServicingWindow] <ServicingWindow> -DaysToRecur <Int32> [-Category <String> ] [-Description <String> ] [-JobVariable <String> ] [-MinutesDuration <Int32> ] [-Name <String> ] [-Owner <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-StartDate <DateTime> ] [-StartTimeOfDay <DateTime> ] [-TimeZone <Int32> ] [ <CommonParameters>]
```

Parameter Set: MonthlyFrequency

```
Set-SCServicingWindow [-ServicingWindow] <ServicingWindow> -DayOfMonth {First | DayTwo | DayThree | DayFour | DayFive | DaySix | DaySeven | DayEight | DayNine | DayTen | DayEleven | DayTwelve | DayThirteen | DayFourteen | DayFifteen | DaySixteen | DaySeventeen | DayEighteen | DayNineteen | DayTwenty | DayTwentyOne | DayTwentyTwo | DayTwentyThree | DayTwentyFour | DayTwentyFive | DayTwentySix | DayTwentySeven | DayTwentyEight | DayTwentyNine | DayThirty | DayThirtyOne | Last} [-Category <String> ] [-Description <String> ] [-JobVariable <String> ] [-MinutesDuration <Int32> ] [-MonthsToRecur <Int32> ] [-Name <String> ] [-Owner <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-StartDate <DateTime> ] [-StartTimeOfDay <DateTime> ] [-TimeZone <Int32> ] [ <CommonParameters>]
```

Parameter Set: MonthlyRelativeFrequency

```
Set-SCServicingWindow [-ServicingWindow] <ServicingWindow> -MonthlyScheduleDayOfWeek {Sunday | Monday | Tuesday | Wednesday | Thursday | Friday | Saturday} -WeekOfMonth {First | Second | Third | Fourth | Last} [-Category <String> ] [-Description <String> ] [-JobVariable <String> ] [-MinutesDuration <Int32> ] [-MonthsToRecur <Int32> ] [-Name <String> ] [-Owner <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-StartDate <DateTime> ] [-StartTimeOfDay <DateTime> ] [-TimeZone <Int32> ] [ <CommonParameters>]
```

Parameter Set: WeeklyFrequency

```
Set-SCServicingWindow [-ServicingWindow] <ServicingWindow> -WeeklyScheduleDayOfWeek <String> [-Category <String> ] [-Description <String> ] [-JobVariable <String> ] [-MinutesDuration <Int32> ] [-Name <String> ] [-Owner <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-StartDate <DateTime> ] [-StartTimeOfDay <DateTime> ] [-TimeZone <Int32> ] [-WeeksToRecur <Int32> ] [ <CommonParameters>]
```



---

## Detailed Description

The Set-SCServicingWindow cmdlet modifies the properties of a servicing window, including the schedule for the servicing window.

## Parameters

### **-Category<String>**

Specifies a category for a servicing window.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-DayOfMonth<DayOfMonthType>**

Specifies the ordinal day of the month on which the schedule starts. For example, 4 indicates the fourth day of the month. "Last" indicates the last day of the month.

The default value is the integer that corresponds to the same day as specified in the StartDate parameter.

Valid integer values: 1 through 31.

Valid string values: "First", "Last"

The acceptable values for this parameter are:

|          |  |
|----------|--|
| First    |  |
| DayTwo   |  |
| DayThree |  |
| DayFour  |  |
| DayFive  |  |
| DaySix   |  |
| DaySeven |  |

|                |  |
|----------------|--|
| DayEight       |  |
| DayNine        |  |
| DayTen         |  |
| DayEleven      |  |
| DayTwelve      |  |
| DayThirteen    |  |
| DayFourteen    |  |
| DayFifteen     |  |
| DaySixteen     |  |
| DaySeventeen   |  |
| DayEighteen    |  |
| DayNineteen    |  |
| DayTwenty      |  |
| DayTwentyOne   |  |
| DayTwentyTwo   |  |
| DayTwentyThree |  |
| DayTwentyFour  |  |
| DayTwentyFive  |  |
| DayTwentySix   |  |
| DayTwentySeven |  |
| DayTwentyEight |  |
| DayTwentyNine  |  |
| DayThirty      |  |
| DayThirtyOne   |  |
| Last           |  |

|                        |       |
|------------------------|-------|
| Aliases                | none  |
| Required?              | true  |
| Position?              | named |
| Default Value          | none  |
| Accept Pipeline Input? | false |

---

|                             |       |
|-----------------------------|-------|
| Accept Wildcard Characters? | false |
|-----------------------------|-------|

### **-DaysToRecur<Int32>**

Specifies, in days, the amount of time between scheduled jobs.

Minimum value: 1

Maximum value: 999

Default value: 1

Example format (to schedule a job to recur every third day): -DaysToRecur 3

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Description<String>**

States a description for the specified object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-MinutesDuration<Int32>**

Specifies, in minutes, a period of time. For example, when used with New-SCServicingWindow or Set-SCServicingWindow, use this parameter to specify the amount of time for which to put a server or service into maintenance mode.

Example format: -DurationMinutes 120

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-MonthlyScheduleDayOfWeek<DayOfWeek>**

Specifies the day of the week to run a job that occurs on a monthly schedule. You can specify only one day of the week. The default value is the current day (if today is Tuesday, Tuesday is the default). Valid values to specify a specific day are: Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday.

Requirement: Use with the parameter WeekOfMonth.

The acceptable values for this parameter are:

|           |  |
|-----------|--|
| Sunday    |  |
| Monday    |  |
| Tuesday   |  |
| Wednesday |  |
| Thursday  |  |

|          |  |
|----------|--|
| Friday   |  |
| Saturday |  |

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-MonthsToRecur<Int32>**

Specifies, in months, the amount of time between scheduled service windows

Minimum value: 1

Maximum value: None

Default value: 1

Example format (to schedule a service window to recur every other month): -MonthsToRecur 2

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Name<String>**

Specifies the name of a VMM object.

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | false |
| Position? | named |

|                             |       |
|-----------------------------|-------|
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Owner<String>**

Specifies the owner of a VMM object in the form of a valid domain user account.

Example format: -Owner "Contoso\ReneeLo"

Example format: -Owner "ReneeLo@Contoso"

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-ServicingWindow<ServicingWindow>**

Specifies a servicing window object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-StartDate<DateTime>**

Specifies the date to start a service window. The default value is the current date. You can type a new date in the short date format for your locale. Or, you can pass a Date-Time object from Get-Date.

Example format that specifies a DateTime object:

```
$StartDate = Get-Date -Year 2012 -Day 5 -Month 4
```

```
-StartDate $StartDate
```

Example formats that specifies a string:

```
-StartDate "August 19, 2011"
```

```
-StartDate "8/19/2011"
```

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-StartTimeOfDay<DateTime>**

Specifies the time of day, or a time-span during a 24-hour period, to start a job or other operation. The default value is the current time.

Example format: -StartTimeOfDay "08:00"

Example format for 2 hrs from 6:00pm: -StartTimeOfDay "18:00-20:00"

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-TimeZone<Int32>**

Specifies a number (an index) that identifies a geographical region that shares the same standard time.

For a list of time zone indexes, see "Microsoft Time Zone Index Values" at:

<http://go.microsoft.com/fwlink/?LinkId=120935>. If no time zone is specified, the default time zone used for a virtual machine is the same time zone setting that is on the virtual machine host.

Example format to specify the GMT Standard Time zone: -TimeZone 085

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-WeeklyScheduleDayOfWeek<String>**

Specifies one or more days of the week to run a job. The default value is the current day of the week.

Valid values to specify an individual day by using a string: Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday. Valid values to specify a set of days in a week: Any set of two or more days separated by commas. Valid values to specify an individual day by using an integer: 1, 2, 3, 4, 5, 6, 7



---

Example format: -WeeklyScheduleDayOfWeek "Monday"

Example format: -WeeklyScheduleDayOfWeek "Monday, Wednesday, Friday"

Example format: -WeeklyScheduleDayOfWeek 5 (to specify a Friday)

Requirement: Use with StartTimeOfDay.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-WeekOfMonth<WeekOfMonthType>**

Specifies a week relative to the first day of the month, such as first, second, third, fourth, or last.

The acceptable values for this parameter are:

|        |  |
|--------|--|
| First  |  |
| Second |  |
| Third  |  |
| Fourth |  |
| Last   |  |

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-WeeksToRecur<Int32>**

Specifies, in weeks, the amount of time between scheduled jobs.

---

Minimum value: 1

Maximum value: None

Default value: 1

Example format (to schedule a job to recur every other week): -WeeksToRecur 2

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **ServicingWindow**

## Examples

### 1: Change the start time of a servicing window.

The first command gets the servicing window object named Backup Staging A and stores the object in the \$SvcWindow variable.

The second command changes the start time of the servicing window stored in \$SvcWindow (Backup Staging A) to 1:00 PM in the GMT Standard time zone.

```
PS C:\> $SvcWindow = Get-SCServicingWindow -Name "Backup Staging A"
```

```
PS C:\> Set-SCServicingWindow -ServicingWindow $SvcWindow -StartTimeOfDay "13:00" -TimeZone 085
```

## Related topics

[Get-SCServicingWindow](#)

[New-SCServicingWindow](#)

[Remove-SCServicingWindow](#)

---

# Set-SCSQLDeployment

---

## Set-SCSQLDeployment

Modifies a SQL Server deployment.

### Syntax

Parameter Set: Default

```
Set-SCSQLDeployment [-SQLDeployment] <SQLDeployment> [-AgentServiceRunAsAccount <VMMCredential> ] [-DeploymentRunAsAccount <VMMCredential> ] [-DeploymentTimeoutSeconds <Int32> ] [-EnableNamedPipes <Boolean> ] [-EnableTCP <Boolean> ] [-InstanceID <String> ] [-InstanceName <String> ] [-JobVariable <String> ] [-MediaSource <String> ] [-MergeSQLAnswerFile <Boolean> ] [-Name <String> ] [-ProductKey <String> ] [-PROTipID <Guid> ] [-ReportingServiceRunAsAccount <VMMCredential> ] [-RunAsynchronously] [-SARunAsAccount <VMMCredential> ] [-SecurityMode <String> ] [-SQLConfigurationFile <Script> ] [-SQLServiceRunAsAccount <VMMCredential> ] [-SQLSysAdminMemberList <String[]> ] [ <CommonParameters>]
```

### Detailed Description

The Set-SCSQLDeployment cmdlet modifies a SQL Server deployment.

### Parameters

#### **-AgentServiceRunAsAccount<VMMCredential>**

Specifies the Run As account to use for the SQL Server agent service.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-DeploymentRunAsAccount<VMMCredential>**

Specifies the Run As account to use for installing SQL Server.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-DeploymentTimeoutSeconds<Int32>**

Specifies the amount of time, in seconds, that the SQL Server deployment waits before timing out.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-EnableNamedPipes<Boolean>**

Indicates that named pipes are used for remote connections.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

### **-EnableTCP<Boolean>**

Indicates that TCP/IP is used for remote connections.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-InstanceID<String>**

Specifies a SQL Server deployment instance ID.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-InstanceName<String>**

Specifies the SQL Server Analysis Services (SSAS) database instance name.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-MediaSource<String>**

Specifies a media source for a SQL Server deployment.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-MergeSQLAnswerFile<Boolean>**

Specifies that the cmdlet merge the specified SQL Server configuration file with the specified guest operating system settings. The default value is TRUE. This parameter is used by the VMM console. You do not need to use this parameter at the command prompt.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-Name<String>**

Specifies the name of a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-ProductKey<String>**

Specifies a product key. The product key is a 25-digit number that identifies the product license. A product key can be used to register VMM or an operating system to be installed on a virtual machine or host.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                        |       |
|------------------------|-------|
| Aliases                | none  |
| Required?              | false |
| Position?              | named |
| Default Value          | none  |
| Accept Pipeline Input? | false |

---

|                             |       |
|-----------------------------|-------|
| Accept Wildcard Characters? | false |
|-----------------------------|-------|

### **-ReportingServiceRunAsAccount<VMMCredential>**

Specifies the Run As account to use for Reporting Services.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-SARunAsAccount<VMMCredential>**

Specifies the Run As account to use for the SQL Server system administrator (SA) password.

|                        |       |
|------------------------|-------|
| Aliases                | none  |
| Required?              | false |
| Position?              | named |
| Default Value          | none  |
| Accept Pipeline Input? | false |



---

|                             |       |
|-----------------------------|-------|
| Accept Wildcard Characters? | false |
|-----------------------------|-------|

### **-SecurityMode<String>**

Specifies the security mode for SQL Server. Valid values are: WindowsAuthentication, SQLServerAuthentication.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-SQLConfigurationFile<Script>**

Specifies a SQL Server configuration file.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-SQLDeployment<SQLDeployment>**

Specifies a SQL Server deployment object.

|               |      |
|---------------|------|
| Aliases       | none |
| Required?     | true |
| Position?     | 1    |
| Default Value | none |

|                             |                |
|-----------------------------|----------------|
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-SQLServiceRunAsAccount<VMMCredential>**

Specifies the Run As account to use for the SQL Server service.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-SQLSysAdminMemberList<String[]>**

Specifies a list of users that are SQL Server administrators.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

### **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **SQLDeployment**

---

## Examples

### 1: Modify an existing SQL Server deployment.

The first command gets the SQL Server profile object named SQLProfile01 and stores the object in the \$SQLProfile variable.

The second command gets the SQL Server deployment object named SQL Deployment from the SQL profile stored in \$SQLProfile, and then stores the object in the \$SQLDeployment variable.

The last command modifies the SQL Server administrators list of the SQL deployment stored in \$SQLDeployment.

```
PS C:\> $SQLProfile = Get-SCSQLProfile -Name "SQLProfile01"
```

```
PS C:\> $SQLDeployment = Get-SCSQLDeployment -SQLProfile $SQLProfile -Name "SQL Deployment 01"
```

```
PS C:\> Set-SCSQLDeployment -SQLDeployment $SQLDeployment -SQLSysAdminMemberList @("Contoso\SQLAdmins","Contoso\User")
```

### Related topics

[Add-SCSQLDeployment](#)

[Get-SCSQLDeployment](#)

[Remove-SCSQLDeployment](#)

---

# Set-SCSQLProfile

---

## Set-SCSQLProfile

Modifies the properties of a SQL Server profile.

### Syntax

Parameter Set: Default

```
Set-SCSQLProfile [-SQLProfile] <SQLProfile> [-Description <String> ] [-JobVariable <String> ] [-Name <String> ] [-Owner <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-Tag <String> ] [-UserRole <UserRole> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Set-SQLProfile cmdlet modifies the properties of a SQL Server profile.

### Parameters

#### **-Description<String>**

States a description for the specified object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Name<String>**

Specifies the name of a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Owner<String>**

Specifies the owner of a VMM object in the form of a valid domain user account.

Example format: -Owner "Contoso\ReneeLo"

Example format: -Owner "ReneeLo@Contoso"

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

---

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-SQLProfile<SQLProfile>**

Specifies a SQL Server profile object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

---

## **-Tag<String>**

Specifies a word or phrase to associate with an object so that you can search for all objects with the specified set of tags. You can search for a subset of tags, or you can search for the full set of tags.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-UserRole<UserRole>**

Specifies a user role object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **SQLProfile**

## Examples

### 1: Change the name of a SQL Server profile and update its description.

The first command gets the SQL Server profile object named SQLProfile01 and stores the object in the \$SQLProfile variable.

The second command renames the SQL Server profile stored in \$SQLProfile to SQLProfile02 and updates its description.

```
PS C:\> $SQLProfile = Get-SCSQLProfile -Name "SQLProfile01"
```

```
PS C:\> Set-SCSQLProfile -SQLProfile $SQLProfile -Name "SQLProfile02" -Description "SQL Profile 02"
```

## Related topics

[Get-SCSQLProfile](#)

[New-SCSQLProfile](#)

[Remove-SCSQLProfile](#)



---

# Set-SCSQLScriptCommand

---

## Set-SCSQLScriptCommand

Modifies the properties of a SQL Server script.

### Syntax

Parameter Set: Default

```
Set-SCSQLScriptCommand [-SQLScriptCommand] <SCSQLScriptCommand> [-CommandParameters <String> ] [-DatabaseName <String> ] [-DeploymentOrder <Int32> ] [-EncryptConnection <Boolean> ] [-ExecutionTimeoutSeconds <Int32> ] [-JobVariable <String> ] [-LoginTimeoutSeconds <Int32> ] [-OutputFilePath <String> ] [-PROTipID <Guid> ] [-RunAsAccount <VMMSCredential> ] [-RunAsynchronously] [-SQLAuthenticationType <String> ] [-SQLScript <Script> ] [-SQLScriptType {PreInstall | PostInstall | PreService | PostService | PreUninstall | PostUninstall} ] [-WarnAndContinueOnError <Boolean> ] [ <CommonParameters> ]
```

### Detailed Description

The Set-SCSQLScriptCommand cmdlet modifies the properties of a SQL Server script associated with an application deployment.

### Parameters

#### **-CommandParameters<String>**

Specifies the parameters for a script or executable program.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-DatabaseName<String>**

Specifies the name of a database for a SQL Server script.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-DeploymentOrder<Int32>**

Specifies the order in which a computer tier, application host, or application is deployed.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-EncryptConnection<Boolean>**

Specifies whether the SQL Server connection is encrypted.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

### **-ExecutionTimeoutSeconds<Int32>**

Specifies the amount of time, in seconds, that the SQL Server script command (SQL statement) will wait before timing out.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-LoginTimeoutSeconds<Int32>**

Specifies the amount of time, in seconds, that a SQL Server login waits before timing out.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-OutputFilePath<String>**

Specifies a file path to store output data from a SQL Server script.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RunAsAccount<VMMSCredential>**

Specifies a Run As account that contains credentials with permission to perform this action.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-SQLAuthenticationType<String>**

Specifies the SQL Server authentication type. Valid values are: SQLServerAuthentication, WindowsAuthentication.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-SQLScript<Script>**

Specifies a SQL Server script.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-SQLScriptCommand<SCSQLScriptCommand>**

Specifies a SQL Server script command object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-SQLScriptType<SQLScriptCommandType>**

Specifies a SQL Server script type. Valid values are: PreInstall, PostInstall, PreService, PostService, PreUninstall, PostUninstall.

The acceptable values for this parameter are:

|               |  |
|---------------|--|
| PreInstall    |  |
| PostInstall   |  |
| PreService    |  |
| PostService   |  |
| PreUninstall  |  |
| PostUninstall |  |

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## -WarnAndContinueOnError<Boolean>

Indicates whether the script should warn the user and continue if the SQL Server script encounters an error while running.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **SQLScriptCommand**

## Examples

### 1: Modify a SQL Server script command.

The first command gets the application profile object named SvcWebAppProfile01 and stores it in the \$AppProfile variable.

The second command gets the application deployment object named SQLDataTierApp01 for the application profile stored in \$ApplicationProfile, and then stores the object in the \$AppDeployment variable.

The third command gets the first PreInstall SQL Server script object (deployment order 1, sql script type PreInstall) associated with the application deployment stored in \$AppDeployment, and then stores the object in the \$SQLScript variable.

The last command modifies the database against which the SQL Server script stored \$SQLScript will run

```
PS C:\> $AppProfile = Get-SCApplicationProfile -Name "SvcWebAppProfile01"
```

```
PS C:\> $AppDeployment = Get-SCApplicationDeployment -ApplicationProfile $AppProfile -Name "SQLDataTierApp01"
```

```
PS C:\> $SQLScript = Get-SCSQLScriptCommand -ApplicationDeployment $AppDeployment | where {$_.DeploymentOrder -eq "1" -and $_.SQLScriptType -eq "PreInstall"}
```

---

```
PS C:\> Set-SCSQLScriptCommand -SQLScriptCommand $SQLScript -DatabaseName "MSOrders"
```

## Related topics

[Add-SCSQLScriptCommand](#)

[Get-SCSQLScriptCommand](#)

[Remove-SCSQLScriptCommand](#)



---

# Set-SCSSHKey

---

## Set-SCSSHKey

Updates the properties of a Linux SSHKey object stored in the VMM library.

### Syntax

Parameter Set: Default

```
Set-SCSSHKey [-SSHKey] <SSHKey> [-Description <String> ] [-Enabled <Boolean> ] [-FamilyName <String> ] [-JobVariable <String> ] [-Name <String> ] [-Owner <String> ] [-PROTipID <Guid> ] [-Release <String> ] [-RunAsynchronously] [-SharePath <String> ] [-UserRole <UserRole> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: EquivalencySet

```
Set-SCSSHKey [-SSHKeys] <List<SSHKey>> -FamilyName <String> -Release <String> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Set-SCSSHKey cmdlet updates the properties of a Linux SSHKey object stored in the VMM Library.

### Parameters

#### **-Description<String>**

States a description for the specified object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

### **-Enabled<Boolean>**

Enables an object when set to \$True, or disables an object when set to \$False. For example, if you want to upgrade software on a virtual machine template, you can disable the template object in the VMM library to temporarily prevent users from using that object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-FamilyName<String>**

Specifies a family name for a physical resource in the VMM library.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-Name<String>**

Specifies the name of a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-Owner<String>**

Specifies the owner of a VMM object in the form of a valid domain user account.

Example format: -Owner "Contoso\ReneeLo"

Example format: -Owner "ReneeLo@Contoso"

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|               |       |
|---------------|-------|
| Aliases       | none  |
| Required?     | false |
| Position?     | named |
| Default Value | none  |

|                             |       |
|-----------------------------|-------|
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Release<String>**

Specifies a string that describes the release of a library resource. VMM automatically creates a release value for every resource imported into the library. After the resource has been imported, the string can be customized.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-SharePath<String>**

Specifies a path to a valid library share on an existing library server that uses a Universal Naming Convention (UNC) path.

Example format: `-SharePath "\\LibServer01\LibShare"`

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-SSHKey<SSHKey>**

Specifies a Linux Administrator SSHKey object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-SSHKeys<List<SSHKey>>**

Specifies a list of Linux Administrator SSHKey objects.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-UserRole<UserRole>**

Specifies a user role object.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **LinuxAdministratorSshKey**

## **Examples**

### **1: Change the description of an SSHKey object.**

The first command gets the SSHKey object named My.sshkey from the library on VMMServer01 and stores the object in the \$sshkey variable.

The second command changes the description of the SSHKey stored in \$sshkey to "My standard SSHKey".

```
PS C:\> $sshkey = Get-SCSSHKey -VMMServer "VMMServer01.Contoso.com" | where {
$_.LibraryServer.Name -eq "LibraryServer01.Contoso.com" -and $_.Name -eq "My.sshkey" }
```

---

```
PS C:\> Set-SCSSHKey -SSHKey $sshkey -Description "My standard SSHKey"
```

## 2: Change the FamilyName property for an SSHKey object.

The first command gets the SSHKey object named My.sshkey and stores the object in the \$SSHKey variable.

The second command sets the FamilyName property to Family01 on the SSHKey object stored in \$SSHKey.

```
PS C:\> $SSHKey = Get-SCSSHKey -Name "My.sshkey"
```

```
PS C:\> Set-SCSSHKey -SSHKey $SSHKey -FamilyName "Family01"
```

## Related topics

[Get-SCSSHKey](#)

[Remove-SCSSHKey](#)

---

# Set-SCStaticIPAddressPool

---

## Set-SCStaticIPAddressPool

Modifies a static IP address pool that is associated with one or more hostgroups.

### Syntax

Parameter Set: Default

```
Set-SCStaticIPAddressPool [-StaticIPAddressPool] <StaticIPAddressPool> [-DefaultGateway <DefaultGateway[]> ] [-Description <String> ] [-DNSSearchSuffix <String[]> ] [-DNSServer <String[]> ] [-DNSSuffix <String> ] [-EnableNetBIOS <Boolean> ] [-IPAddressRangeEnd <String> ] [-IPAddressRangeStart <String> ] [-IPAddressReservedSet <String> ] [-JobVariable <String> ] [-Name <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VIPAddressSet <String> ] [-VMMServer <ServerConnection> ] [-WINSSEServer <String[]> ] [ <CommonParameters>]
```

### Detailed Description

The Set-SCStaticIPAddressPool cmdlet modifies a Virtual Machine Manager (VMM) static IP address pool. A static IP address pool can be associated with one or more logical network definitions.

### Parameters

#### **-DefaultGateway<DefaultGateway[]>**

Specifies a default gateway object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |



---

## **-Description<String>**

States a description for the specified object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-DNSSearchSuffix<String[]>**

Specifies one or more strings that are appended to a host name to resolve a DNS address.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-DNSServer<String[]>**

Specifies the IP address of one or more DNS servers. Valid formats: IPv4 or IPv6 address.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

### **-DNSSuffix<String>**

Specifies the default DNS suffix associated with a NIC.

Example format: -DNSSuffix "Domain01.Contoso.com"

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-EnableNetBIOS<Boolean>**

Indicates whether NetBIOS over TCP/IP is enabled for a NIC.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-IPAddressRangeEnd<String>**

Specifies the last IP address in a range of IP addresses. Use with IPAddressRangeStart.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-IPAddressRangeStart<String>**

Specifies the first IP address in a range of IP addresses. Use with IPAddressRangeEnd.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-IPAddressReservedSet<String>**

Specifies a set of IP addresses within an IP subnet that is reserved for other use.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-Name<String>**

Specifies the name of a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-StaticIPAddressPool<StaticIPAddressPool>**

Specifies an IP address pool from which you can assign static IP addresses.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-VIPAddressSet<String>**

Specifies a set of IP addresses within an IP subnet that is reserved for configuring virtual IPs (VIPs) in load balancers.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

---

## -WINS Server<String[]>

Specifies the IP address of one or more WINS servers. Valid formats: IPv4 or IPv6 address.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **StaticIPAddressPool**

## Examples

### 1: Add DNS Servers to a static IP address pool.

The first command gets the host group with the path of All Hosts\HostGroup02\Production and stores it in the \$HostGroup variable.

The second command gets the static address pool named Production IP Address Pool for the host group stored in the \$HostGroup variable using the IPv4 address for the specified subnet.

The third command gets the DNSServer for the IP address pool stored in the \$IPPool variable, and then the fifth command adds an address to the address array stored in the \$DNSServerIPAddress variable.

The last command updates the IP address pool stored in \$IPPool with the address array stored in \$DNSServerIPAddress.

```
PS C:\> $HostGroup = Get-SCVMHostGroup | where { $_.Path -eq "All
Hosts\HostGroup02\Production" }
PS C:\> $IPPool = Get-SCStaticIPAddressPool -IPv4 -Subnet "10.0.0.0/24" -VMHostGroup
$HostGroup -Name "Production IP Address Pool"
PS C:\> $DNSServerIPAddress = $IPPool.DNSServers
PS C:\> $DNSServerIPAddress += "10.0.0.1"
```

---

```
PS C:\> Set-SCStaticIPAddressPool -StaticIPAddressPool $IPPool -DNSServer  
$DNSServerIPAddress
```

## Related topics

[Get-SCStaticIPAddressPool](#)

[New-SCStaticIPAddressPool](#)

[Remove-SCStaticIPAddressPool](#)

---

# Set-SCStorageArray

---

## Set-SCStorageArray

Modifies the properties of a storage array object.

### Syntax

Parameter Set: Default

```
Set-SCStorageArray [-StorageArray] <StorageArray> [-CreateStorageGroupsPerCluster <Boolean> ] [-Description <String> ] [-JobVariable <String> ] [-LogicalUnitCopyMethod {Snapshot | Clone} ] [-MaximumStorageLogicalUnitNameLength <Int16> ] [-Name <String> ] [-PROTipID <Guid> ] [-RemoveStoragePoolFromManagement <StoragePool[]> ] [-RunAsynchronously] [ <CommonParameters>]
```

Parameter Set: EnablePoolManagement

```
Set-SCStorageArray [-StorageArray] <StorageArray> -AddStoragePoolToManagement <StoragePool[]> -StorageClassificationAssociation <StorageClassification[]> [-CreateStorageGroupsPerCluster <Boolean> ] [-Description <String> ] [-JobVariable <String> ] [-LogicalUnitCopyMethod {Snapshot | Clone} ] [-MaximumStorageLogicalUnitNameLength <Int16> ] [-Name <String> ] [-PROTipID <Guid> ] [-RemoveStoragePoolFromManagement <StoragePool[]> ] [-RunAsynchronously] [ <CommonParameters>]
```

### Detailed Description

The Set-SCStorageArray cmdlet modifies the properties of a Virtual Machine Manager (VMM) storage array object.

### Parameters

#### **-AddStoragePoolToManagement<StoragePool[]>**

Enables management of a storage pool through VMM. When set to \$True, VMM imports all logical unit objects hosted by the storage pool.

|               |       |
|---------------|-------|
| Aliases       | none  |
| Required?     | true  |
| Position?     | named |
| Default Value | none  |



---

|                             |       |
|-----------------------------|-------|
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-CreateStorageGroupsPerCluster<Boolean>**

Indicates whether a storage group is created for each cluster.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Description<String>**

States a description for the specified object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|               |       |
|---------------|-------|
| Aliases       | none  |
| Required?     | false |
| Position?     | named |
| Default Value | none  |

|                             |       |
|-----------------------------|-------|
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-LogicalUnitCopyMethod<StorageLogicalUnitCopyMethod>**

Specifies the method used by the array to copy an existing logical unit. Valid values: Clone, Snapshot.

The acceptable values for this parameter are:

|          |  |
|----------|--|
| Snapshot |  |
| Clone    |  |

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-MaximumStorageLogicalUnitNameLength<Int16>**

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Name<String>**

Specifies the name of a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RemoveStoragePoolFromManagement<StoragePool[]>**

Removes a storage pool from VMM management. This parameter deletes all logical unit information from VMM, but does not delete any data from the logical units themselves.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-StorageArray<StorageArray>**

Specifies a storage array object, This can be a Fibre Channel or iSCSI storage sub-system that is used to store virtual machine configuration and virtual disks.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-StorageClassificationAssociation<StorageClassification[]>**

Specifies an array of storage classification objects that is associated with a storage pool.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **StorageArray**

## Examples

### 1: Change the name of a storage array

The first command gets the first item in the storage array and stores it in the \$Array variable.

The second command changes the name of the storage array stored in the \$Array variable to "New Name".

```
PS C:\> $Array = @(Get-SCStorageArray)[0]
```

```
PS C:\> Set-SCStorageArray -StorageArray $Array -Name "New Name"
```

## Related topics

[Get-SCStorageArray](#)

---

# Set-SCStorageClassification

---

## Set-SCStorageClassification

Modifies the properties of an existing storage classification.

### Syntax

Parameter Set: Default

```
Set-SCStorageClassification [-StorageClassification] <StorageClassification> [-Description <String> ] [-JobVariable <String> ] [-Name <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

### Detailed Description

The Set-SCStorageClassification cmdlet modifies the properties of an existing storage classification.

### Parameters

#### -Description<String>

States a description for the specified object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### -JobVariable<String>

Specifies that job progress is tracked and stored in the variable named by this parameter.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Name<String>**

Specifies the name of a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-StorageClassification<StorageClassification>**

Specifies a storage classification object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **StorageClassification**

## **Examples**

### **1: Update the name of a storage classification.**

The first command gets the first item in the storage classification array and stores it in the \$Class variable.

The second command changes the name of the storage classification stored in the \$Class variable to Tier2.

```
PS C:\> $Class = @(Get-SCStorageClassification)[0]
```

```
PS C:\> Set-SCStorageClassification -StorageClassification $Class -Name "Tier2"
```



---

## Related topics

[Get-SCStorageClassification](#)

[New-SCStorageClassification](#)

[Remove-SCStorageClassification](#)

---

# Set-SCStorageFileServer

---

## Set-SCStorageFileServer

Adds or removes a storage file share from VMM management.

### Syntax

Parameter Set: Default

```
Set-SCStorageFileServer [-StorageFileServer] <StorageFileServer> [-Description <String> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RemoveStorageFileShareFromManagement <StorageFileShare[]> ] [-RunAsynchronously] [ <CommonParameters>]
```

Parameter Set: EnableFileShareManagement

```
Set-SCStorageFileServer [-StorageFileServer] <StorageFileServer> -AddStorageFileShareToManagement <StorageFileShare[]> [-Description <String> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RemoveStorageFileShareFromManagement <StorageFileShare[]> ] [-RunAsynchronously] [ <CommonParameters>]
```

### Detailed Description

The Set-SCStorageFileServer adds or removes a storage file share from Virtual Machine Manager (VMM) management.

### Parameters

#### **-AddStorageFileShareToManagement<StorageFileShare[]>**

Enables management of the specified storage file share through VMM.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-Description<String>**

States a description for the specified object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-RemoveStorageFileShareFromManagement<StorageFileShare[]>**

Removes the specified storage file share from VMM management.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-StorageFileServer<StorageFileServer>**

Specifies a storage file server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Examples

### 1: Add a storage file share to VMM management.

The first command gets the storage file server object named FileServer01 and stores the object in the \$FileServer variable.

The second command gets the storage file share object named FileShare01 and stores the object in the \$FileShare variable.

The last command brings FileShare01 under VMM management.

```
PS C:\> $FileServer = Get-SCStorageFileServer -Name "FileServer01.Contoso.com"
PS C:\> $FileShare = Get-SCStorageFileShare -Name "FileShare01"
PS C:\> Set-SCStorageFileServer -StorageFileServer $FileServer -
AddStorageFileShareToManagement $FileShare
```

### 2: Remove a storage file share from VMM management.

The first command gets the storage file server object named FileServer01 and stores the object in the \$FileServer variable.

The second command gets the storage file share object named FileShare01 and stores the object in the \$FileShare variable.

The last command removes FileShare01 from VMM management.

```
PS C:\> $FileServer = Get-SCStorageFileServer -Name "FileServer01.Contoso.com"
PS C:\> $FileShare = Get-SCStorageFileShare -Name "FileShare01"
PS C:\> Set-SCStorageFileServer -StorageFileServer $FileServer -
RemoveStorageFileShareFromManagement $FileShare
```

## Related topics

[Get-SCStorageFileServer](#)

[Get-SCStorageFileShare](#)

---

# Set-SCStorageFileShare

---

## Set-SCStorageFileShare

Modifies a storage file share.

### Syntax

Parameter Set: Default

```
Set-SCStorageFileShare [-StorageFileShare] <StorageFileShare> [-Description <String> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-StorageClassification <StorageClassification> ] [ <CommonParameters>]
```

### Detailed Description

The Set-SCStorageFileShare cmdlet modifies a storage file share.

### Parameters

#### **-Description<String>**

States a description for the specified object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-StorageClassification<StorageClassification>**

Specifies a storage classification object.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-StorageFileShare<StorageFileShare>**

Specifies a storage file share.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## **Examples**

### **1: Update the storage classification for a storage file share.**

The first command gets the storage file share object named FileShare01 and stores the object in the \$FileShare variable.

The second command gets the storage classification object named StorageClassification02 and stores the object in the \$StorageClass variable.

The last command changes the storage classification on FileShare01 to StorageClassification02.

```
PS C:\> $FileShare = Get-SCStorageFileShare -Name "FileShare01"
```

```
PS C:\> $StorageClass = Get-SCStorageClassification -Name "StorageClassification02"
```

```
PS C:\> Set-SCStorageFileShare -StorageFileShare $FileShare -StorageClassification $StorageClass
```



---

## Related topics

[Get-SCStorageFileShare](#)

[New-SCStorageFileShare](#)

[Register-SCStorageFileShare](#)

[Remove-SCStorageFileShare](#)

[Repair-SCStorageFileShare](#)

[Unregister-SCStorageFileShare](#)

---

# Set-SCStorageLogicalUnit

---

## Set-SCStorageLogicalUnit

Updates the metadata of a storage logical unit object.

### Syntax

Parameter Set: Default

```
Set-SCStorageLogicalUnit [-StorageLogicalUnit] <StorageLogicalUnit> [-Description <String> ]  
[-JobVariable <String> ] [-LogicalUnitCopySource <StorageLogicalUnit> ] [-Name <String> ] [-  
PROTipID <Guid> ] [-RunAsynchronously] [-VMHostGroup <HostGroup> ] [-WorkloadType {Shared |  
Dedicated} ] [ <CommonParameters>]
```

### Detailed Description

The Set-SCStorageLogicalUnit cmdlet updates the metadata of a storage logical unit object. Set-SCStorageLogicalUnit does not modify the data on the logical unit itself.

### Parameters

#### **-Description<String>**

States a description for the specified object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

---

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-LogicalUnitCopySource<StorageLogicalUnit>**

Specifies a storage logical unit from which a clone is copied.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Name<String>**

Specifies the name of a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-StorageLogicalUnit<StorageLogicalUnit>**

Specifies a storage logical unit object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-VMHostGroup<HostGroup>**

Specifies a virtual machine host group object or an array of host group objects.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-WorkloadType<StorageLogicalUnitWorkloadType>**

Specifies a storage logical unit workload type. Valid values are: Shared, Dedicated.

The acceptable values for this parameter are:

|           |  |
|-----------|--|
| Shared    |  |
| Dedicated |  |

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Examples**

### **1: Update the name of a storage logical unit.**

The first command gets the logical unit object named LUN01 and stores the object in the \$LogicalUnit variable.

---

The second command changes the name of the logical unit object stored in \$LogicalUnit to "New Name for Logical Unit".

```
PS C:\> $LogicalUnit = Get-SCStorageLogicalUnit -Name "LUN01"
```

```
PS C:\> Set-SCStorageLogicalUnit -StorageLogicalUnit $LogicalUnit -Name "New Name for Logical Unit"
```

## 2: Allocate storage to a host group.

The first command gets the storage logical unit object named LUN01 and stores the object in the \$LU variable

The second command gets the host group object named All Hosts and stores the object in the \$HostGroup variable.

The last command allocates LUN01 to host group All Hosts.

```
PS C:\> $LU = Get-SCStorageLogicalUnit -Name "LUN01"
```

```
PS C:\> $HostGroup = Get-SCVMHostGroup -Name "All Hosts"
```

```
PS C:\> Set-SCStorageLogicalUnit -StorageLogicalUnit $LU -VMHostGroup $HostGroup
```

## Related topics

[Get-SCStorageLogicalUnit](#)

[New-SCStorageLogicalUnit](#)

[Register-SCStorageLogicalUnit](#)

[Remove-SCStorageLogicalUnit](#)

[Unregister-SCStorageLogicalUnit](#)

---

# Set-SCStoragePool

---

## Set-SCStoragePool

Modifies a storage pool object in the VMM database.

### Syntax

Parameter Set: Default

```
Set-SCStoragePool [-StoragePool] <StoragePool> [-AddVMHostGroup <HostGroup[]> ] [-Description <String> ] [-JobVariable <String> ] [-Name <String> ] [-PROTipID <Guid> ] [-RemoveVMHostGroup <HostGroup[]> ] [-RunAsynchronously] [-StorageClassification <StorageClassification> ] [ <CommonParameters> ]
```

### Detailed Description

The Set-SCStoragePool cmdlet modifies a storage pool object in the Virtual Machine Manager (VMM) database. You can change the properties of a storage pool, add a VMHostGroup to the storage pool, or remove a VMHostGroup from a storage pool.

### Parameters

#### **-AddVMHostGroup<HostGroup[]>**

Adds one or more host groups to an existing host group array or private cloud.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### **-Description<String>**

States a description for the specified object.

---

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Name<String>**

Specifies the name of a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.



|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RemoveVMHostGroup<HostGroup[]>**

Removes one or more host groups from a host group array or private cloud.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-StorageClassification<StorageClassification>**

Specifies a storage classification object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-StoragePool<StoragePool>**

Specifies a storage pool object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **StoragePool**

## **Examples**

### **1: Change the name of a storage pool.**

The first command gets all storage pool objects and places them in an array. The command then stores the first item in the storage pool array in the \$Pool variable.

The second command changes the name of the storage pool stored in the \$Pool variable to "New name of pool".

---

```
PS C:\> $Pool = @(Get-SCStoragePool)[0]
PS C:\> Set-SCStoragePool -StoragePool $Pool -Name "New name of pool"
```

## 2: Set the classification for a storage pool.

The first command gets the storage pool object with the ID of 346e17e9-d50a-480e-8dec-c41d7e2125b0 and stores the object in the \$Pool variable.

The second command gets the storage classification object named StorageClassification01 and stores the object in the \$Classification variable.

The last command associates the storage classification stored in \$Classification (StorageClassification01) with the storage pool stored in \$Pool.

```
PS C:\> $Pool = Get-SCStoragePool -ID "346e17e9-d50a-480e-8dec-c41d7e2125b0"
PS C:\> $Classification = Get-SCStorageClassification -Name "StorageClassification01"
PS C:\> Set-SCStoragePool -StoragePool $Pool -StorageClassification $Classification
```

## Related topics

[Get-SCStoragePool](#)

---

# Set-SCStorageProvider

---

## Set-SCStorageProvider

Modifies a storage provider object in VMM.

### Syntax

Parameter Set: Default

```
Set-SCStorageProvider [-StorageProvider] <StorageProvider> [-Certificate <ClientCertificate> ] [-Description <String> ] [-JobVariable <String> ] [-Name <String> ] [-NetworkDeviceName <String> ] [-PROTipID <Guid> ] [-RunAsAccount <RunAsAccount> ] [-RunAsynchronously] [-TCPPort <UInt32> ] [ <CommonParameters>]
```

### Detailed Description

The Set-SCStorageProvider cmdlet modifies a storage provider object in Virtual Machine Manager (VMM).

### Parameters

#### **-Certificate<ClientCertificate>**

Specifies a security certificate object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### **-Description<String>**

States a description for the specified object.

---

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Name<String>**

Specifies the name of a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-NetworkDeviceName<String>**

Specifies the name of a network device.

---

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsAccount<RunAsAccount>**

Specifies a Run As account that contains credentials with permission to perform this action.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-StorageProvider<StorageProvider>**

Specifies a storage provider object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-TCPPort<UInt32>**

Specifies a numeric value that represents a TCP port.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

---

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **StorageProvider**

## Examples

### 1: Change the name of a storage provider.

The first command gets the storage provider named StorProv01 and stores it in the \$Provider variable.

The second command gets RunAs account RunAsAccount01 and stores it in the \$RunAsAcct variable.

The last command sets the network device name of the storage provider stored in the \$Provider variable to NewStorProvName using the RunAs account stored in \$RunAsAcct.

```
PS C:\> $Provider = Get-SCStorageProvider -Name "StorProv01.Contoso.com"
```

```
PS C:\> $RunAsAcct = Get-SCRunAsAccount -Name "RunAsAccount01"
```

```
PS C:\> Set-SCStorageProvider -StorageProvider $Provider -NetworkDeviceName  
"http://StorProv01.Contoso.com" -Name "NewStorProvName.Contoso.com" -RunAsAccount $RunAsAcct
```

### 2: Change the TCP/IP port of a storage provider.

The first command gets the storage provider named StorProv01 and stores it in the \$Provider variable.

The second command gets the RunAs account named RunAsAccount01 and stores the object in the \$RunAsAcct variable.

The third command changes the TCP/IP port of the storage provider stored in \$Provider to 40441 using the RunAs account stored in \$RunAsAcct.

```
PS C:\> $Provider = Get-SCStorageProvider -Name "StorProv01.Contoso.com"
```

```
PS C:\> $RunAsAcct = Get-SCRunAsAccount -Name "RunAsAccount01"
```

```
PS C:\> Set-SCStorageProvider -StorageProvider $Provider -TCPport 40441 -RunAsAccount  
$RunAsAcct
```

## Related topics

[Add-SCStorageProvider](#)

[Get-SCStorageProvider](#)

[Read-SCStorageProvider](#)

[Remove-SCStorageProvider](#)



---

# Set-SCStorageVolume

---

## Set-SCStorageVolume

Modifies the setting for a volume on a host that enables VMM to evaluate that volume as available storage during the virtual machine placement process.

### Syntax

Parameter Set: Default

```
Set-SCStorageVolume [-StorageVolume] <StorageVolume> -AvailableForPlacement <Boolean> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

### Detailed Description

The Set-SCStorageVolume cmdlet modifies the setting that determines whether Virtual Machine Manager (VMM) will evaluate a specific volume on a host server as available storage during the virtual machine placement process.

During the placement process, VMM evaluates managed hosts, including the volumes on those managed hosts, when calculating a recommendation for the best location on which to deploy a virtual machine. If you specify that a volume on the host will not be included when VMM performs its automatic placement calculation, you can still choose to manually deploy a virtual machine on that volume.

### Parameters

#### **-AvailableForPlacement<Boolean>**

Indicates whether the VMM placement process will consider this host or this volume on a host to be eligible as a possible location on which to deploy virtual machines. If this parameter is set to False, you can choose to deploy virtual machines on this host or volume anyway. The default value is True. This parameter does not apply to VMware ESX hosts.

When this parameter is used with network adapters, if set to \$False, then placement will not consider the logical networks configured on this network adapter to determine if the host is suitable for connecting a virtual machine.

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | true  |
| Position? | named |

---

|                             |       |
|-----------------------------|-------|
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | false |
| Position? | named |

|                             |       |
|-----------------------------|-------|
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-StorageVolume<StorageVolume>**

Specifies a storage volume object on a specific virtual machine host.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **StorageVolume**

## **Examples**

### **1: Make a volume on a host available for placement.**

The first command gets the host object named VMHost01 and stores the object in the \$VMHost variable.

The second command gets the storage volume object for the host stored in \$VMHost and then stores the object in the \$StorageVol variable. This example assumes that VMHost01 has only one volume.

The last command makes the first volume object on VMHost01 available for placement. Setting the parameter AvailableForPlacement to TRUE enables the VMM placement process to evaluate this volume on VMHost01 as a possible candidate to host virtual machines.

```
PS C:\> $VMHost = Get-SCVMHost -ComputerName "VMHost01.Contoso.com"
```

```
PS C:\> $StorageVol = Get-SCStorageVolume -VMHost $VMHost
```

---

```
PS C:\> Set-SCStorageVolume -StorageVolume $StorageVol[0] -AvailableForPlacement $True
```

## 2: Make a second volume on a host available for placement.

The first command gets the host object named VMHost02 and stores the object in the \$VMHost variable.

The second command gets all storage volume objects VMHost02 and stores the objects in the object array named \$StorageVols. This example assumes that VMHost02 has at least two volumes.

The last command makes the second volume stored in the \$StorageVols array available for placement.

```
PS C:\> $VMHost = Get-SCVMHost -ComputerName "VMHost02.Contoso.com"
```

```
PS C:\> $StorageVols = Get-SCStorageVolume -VMHost $VMHost
```

```
PS C:\> Set-SCStorageVolume -StorageVolume $StorageVols[1] -AvailableForPlacement $True
```

## Related topics

[Get-SCStorageVolume](#)

---

# Set-SCUpdate

---

## Set-SCUpdate

Accepts Microsoft Software License Terms for software updates that require acceptance.

### Syntax

Parameter Set: Default

```
Set-SCUpdate [-Update] <SoftwareUpdate> -AcceptLicenseAgreement [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: UIOnlyEULAAcceptance

```
Set-SCUpdate [-Update] <SoftwareUpdate> -AcceptLicenseAgreement -ClientID <Guid> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Set-Update cmdlet is used to accept Microsoft Software License Terms for updates that require acceptance.

### Parameters

#### -AcceptLicenseAgreement

Indicates that the Microsoft Software License Terms for a software update are accepted.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-ClientID<Guid>**

For internal use only (not for use in your code).

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-Update<SoftwareUpdate>**

Specifies a software update object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- Update

## Examples

### 1: Accept the license agreement for an update.

The first command gets the update for KB article 948465 and stores the object in the \$Update variable.

The second command displays the license agreement text for the update stored in \$Update and prompts you to accept or decline it.

```
PS C:\> $Update = Get-SCUpdate -KBArticle "948465"
```

```
PS C:\> Set-SCUpdate -Update $Update -AcceptLicenseAgreement
```

## Related topics

[Get-SCUpdate](#)



---

# Set-SCUpdateServer

---

## Set-SCUpdateServer

Configures the settings of a Windows Server Update Services computer that has been added to VMM.

### Syntax

Parameter Set: Default

```
Set-SCUpdateServer [-UpdateServer] <UpdateServer> [-JobVariable <String> ] [-PROTipID <Guid> ] [-ProxyCredential <PSCredential> ] [-RunAsynchronously] [-UpdateCategories <List<String>> ] [-UpdateClassifications <List<String>> ] [-UpdateLanguages <List<String>> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ConfigSettingsChange

```
Set-SCUpdateServer [-UpdateServer] <UpdateServer> -AllowConfigurationChanges <Boolean> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: DisableProxy

```
Set-SCUpdateServer [-UpdateServer] <UpdateServer> -DisableProxy [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-UpdateCategories <List<String>> ] [-UpdateClassifications <List<String>> ] [-UpdateLanguages <List<String>> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: EnableProxy

```
Set-SCUpdateServer [-UpdateServer] <UpdateServer> -EnableProxy -IsProxyAccessAnonymous <Boolean> -ProxyServerName <String> -ProxyServerPort <UInt32> [-JobVariable <String> ] [-PROTipID <Guid> ] [-ProxyCredential <PSCredential> ] [-RunAsynchronously] [-UpdateCategories <List<String>> ] [-UpdateClassifications <List<String>> ] [-UpdateLanguages <List<String>> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ManualSync

```
Set-SCUpdateServer [-UpdateServer] <UpdateServer> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-UpdateCategories <List<String>> ] [-UpdateClassifications <List<String>> ] [-UpdateLanguages <List<String>> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Set-SCUpdate Server cmdlet configures the settings of a Microsoft Windows Server Update Services (WSUS) computer that has been added to Virtual Machine Manager (VMM). Configuration settings that can be modified include: proxy server, proxy port, proxy credentials, languages, product categories, and classifications.

---

For more information about how to add a WSUS computer to VMM, type: "Get-Help Add-SCUpdateServer".

## Parameters

### **-AllowConfigurationChanges<Boolean>**

Indicates whether configuration changes to the update server are allowed.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-DisableProxy**

Disables the update server from using a proxy server when synchronizing updates.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-EnableProxy**

Enables the update server to use a proxy server when synchronizing updates.

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | true  |
| Position? | named |

---

|                             |       |
|-----------------------------|-------|
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-IsProxyAccessAnonymous<Boolean>**

Indicates whether the update server requires a proxy server to connect to the Internet.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | false |
| Position? | named |

|                             |       |
|-----------------------------|-------|
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-ProxyCredential<PSCredential>**

Specifies a credential object that contains the user name and password of an account that has permission to communicate with the proxy server.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-ProxyServerName<String>**

Specifies the name of the proxy server that the update server will use to connect with the Microsoft Update Catalog on the Internet.

Example format: `-ProxyServerName "proxy.contoso.com"`

NOTE: Special characters (such as a forward slash "/") cannot be used with this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-ProxyServerPort<UInt32>**

Specifies the port that the update server will use to communicate with the specified proxy server.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-UpdateCategories<List<String>>**

Specifies one or more products that the update server will synchronize.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-UpdateClassifications<List<String>>**

Specifies one or more update classifications that the update server will synchronize. Valid values are: Applications, Critical Updates, Definition Updates, Drivers, Feature Packs, Security Updates, Service Packs, Tools, Update Rollups, Updates.

---

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-UpdateLanguages<List<String>>**

Specifies one or more supported update languages that the update server will synchronize.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-UpdateServer<UpdateServer>**

Specifies a VMM update server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

---

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **UpdateServer**

## Examples

### 1: Enable the proxy setting on an update server.

The first command gets the update server object named WSUSComputer01 and stores the object in the \$UpdateServer variable.

The second command creates an array named \$Categories and populates the array with two products.

The third command creates an array named \$Classification and populates the array with two update classifications.

The fourth command creates an array named \$Languages and populates the array with two supported update languages, English and Brazilian Portuguese.

The last command enables the proxy setting for the update server stored in \$UpdateServer, sets the proxy address and port, and sets the update categories, classifications, and languages to synchronize.

```
PS C:\> $UpdateServer = Get-SCUpdateServer -ComputerName "WSUSComputer01"
```

```
PS C:\> $Categories = @("SQL Server", "Windows")
```

```
PS C:\> $Classifications = @("Security Updates", "Service Packs")
```

```
PS C:\> $Languages = @("en", "pt-br")
```

```
PS C:\> Set-SCUpdateServer -UpdateServer $UpdateServer -EnableProxy -ProxyServerName  
"proxy.contoso.com" -ProxyServerPort "88" -IsProxyAccessAnonymous $True -UpdateCategories  
$Categories -UpdateClassifications $Classifications -UpdateLanguages $Languages
```

---

## 2: Specify a new product type setting for an update server.

The first command gets the update server object named WSUSComputer01 and stores the object in the \$UpdateServer variable.

The second command enables the service packs update classification for the update server stored in \$UpdateServer.

```
PS C:\> $UpdateServer = Get-SCUpdateServer -ComputerName "WSUSComputer01"
```

```
PS C:\> Set-SCUpdateServer -UpdateServer $UpdateServer -UpdateClassifications "Service Packs"
```

### Related topics

[Add-SCUpdateServer](#)

[Get-SCUpdateServer](#)

[Remove-SCUpdateServer](#)



---

# Set-SCUplinkPortProfileSet

---

## Set-SCUplinkPortProfileSet

Sets an uplink port profile set object.

### Syntax

Parameter Set: Default

```
Set-SCUplinkPortProfileSet [-UplinkPortProfileSet] <UplinkPortProfileSet> [[-Description] <String> ] [-AddExtensionUplinkPortProfiles <ExtensionUplinkPortProfile[]> ] [-JobGroup <Guid> ] [-JobVariable <String> ] [-LogicalSwitch <LogicalSwitch> ] [-Name <String> ] [-NativeUplinkPortProfile <NativeUplinkPortProfile> ] [-PROTipID <Guid> ] [-RemoveExtensionUplinkPortProfiles <ExtensionUplinkPortProfile[]> ] [-RemoveNativeUplinkPortProfile] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [<CommonParameters>]
```

### Detailed Description

The Set-SCUplinkPortProfileSet cmdlet updates the properties of an uplink port profile set object.

### Parameters

#### **-AddExtensionUplinkPortProfiles<ExtensionUplinkPortProfile[]>**

Adds the specified array of uplink port profile objects.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

---

## **-Description<String>**

States a description for the specified object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | 2     |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-JobGroup<Guid>**

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-LogicalSwitch<LogicalSwitch>**

Specifies a logical switch object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-Name<String>**

Specifies the name of a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-NativeUplinkPortProfile<NativeUplinkPortProfile>**

Specifies a native uplink port profile object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

---

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RemoveExtensionUplinkPortProfiles<ExtensionUplinkPortProfile[]>**

Removes the specified array of extension uplink port profiles.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-RemoveNativeUplinkPortProfile**

Indicates that the native uplink port profile is removed.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-UplinkPortProfileSet<UplinkPortProfileSet>**

Specifies an uplink port profile set object.

To obtain an uplink port profile set object, use the Get-SCUplinkPortProfileSet cmdlet.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Examples

### 1: Remove a native uplink port profile from an uplink port profile set.

This command gets the uplink port profile set object named UplinkPortProfileSet01 and uses the pipeline operator to pass the object to the Set-SCUplinkPortProfileSet which removes the native uplink port profile from the uplink port profile set object.

```
PS C:\> Get-SCUplinkPortProfileSet -Name "UplinkPortProfileSet01" | Set-SCUplinkPortProfileSet -RemoveNativeUplinkPortProfile
```

## Related topics

[Get-SCUplinkPortProfileSet](#)

[New-SCUplinkPortProfileSet](#)

[Remove-SCUplinkPortProfileSet](#)

---

# Set-SCUserRole

---

## Set-SCUserRole

Modifies the settings for an existing VMM user role.

### Syntax

Parameter Set: Default

```
Set-SCUserRole [-AddMember <String[]> ] [-AddScope <ClientObject[]> ] [-Description <String> ] [-JobGroup <Guid> ] [-JobVariable <String> ] [-Name <String> ] [-Permission {Create | PauseAndResume | Start | Stop | AllowLocalAdmin | RemoteConnect | Remove | Shutdown | Checkpoint | Store | Save | Author | CanShare | CanReceive | CreateFromVHDOOrTemplate | CheckpointRestoreOnly | AuthorVMNetwork} ] [-PROTipID <Guid> ] [-RemoveLibraryStoreSharePath] [-RemoveMember <String[]> ] [-RemoveScope <ClientObject[]> ] [-RemoveVMNetworkMaximum] [-RemoveVMNetworkMaximumPerUser] [-RunAsynchronously] [-ShowPROTips <Boolean> ] [-UserRole <UserRole> ] [-UserRoleDataPath <String> ] [-VMMServer <ServerConnection> ] [-VMNetworkMaximum <UInt16> ] [-VMNetworkMaximumPerUser <UInt16> ] [ <CommonParameters> ]
```

### Detailed Description

The Set-SCUserRole cmdlet modifies the settings for an existing Virtual Machine Manager (VMM) user role. The settings that you can modify depend on the type of VMM user role.

VMM ADMINISTRATOR (Administrator)

-----

You can add members to or remove members from the Administrator user role, but you cannot limit the scope of objects that members of this role can manage.

DELEGATED ADMINISTRATOR (DelegatedAdmin)

-----

You can add members to and remove members from, and you can expand or restrict the scope of a Delegated Administrator user role. You can grant members of this user role permission to manage all of the objects in one or more private clouds and host groups and/or allow users to manage all of the objects stored on one or more library servers. Within that framework, you cannot limit the actions that members of the Delegated Administrator user role can perform.

READ-ONLY ADMINISTRATOR (ReadOnlyAdmin)

-----

You can add members to and remove members from, and you can expand or restrict the scope of a Read-Only Administrator User role. However, the members of the user role can only view the

---

properties, status, and job status of the objects within their assigned scope; they cannot modify any of the objects.

#### SELF-SERVICE USER (SelfServiceUser)

-----

You can add members to or remove members from, and you can expand or limit the scope and actions of members of a Self-Service User role. You can grant members of a self-service user role permission to manage all of the objects in one or more private clouds; permission to create virtual machines; permission to store virtual machines in the stored virtual machine path in the cloud that the virtual machine is on; and permission to use one or more template objects to create virtual machines. Within that framework, you can grant members of a Self-Service User role one or more actions that self-service users can take. You can also limit the number of virtual machines that self-service users can create by setting a quota that applies to each user or to all users collectively.

The actions that you can grant a Self-Service user include the following:

#### ActionDescription

-----

AllowLocalAdminGrants user local administrator rights on virtual machines

AuthorAuthor virtual machine and service templates

CanShareShare resources with other Self-Service users

CanReceiveReceive resources from other Self-Service users

CheckpointCreate and manage virtual machine checkpoints

CheckpointRestoreOnly Can only restore a checkpoint

CreateCreate virtual machines and services from templates only

CreateFromVHDOrTemplate Create virtual machines and services from VHD files or templates

PauseAndResumePause and resume virtual machines and services

RemoteConnectRemotely connect to virtual machines

RemoveRemove virtual machines and services

SaveSave virtual machines and services

ShutdownShut down virtual machines

StartStart virtual machines and services

StopStop virtual machines and services

StoreStore virtual machines in a library

## Parameters

### **-AddMember<String[]>**

Adds one or more members to an object that has the concept of members, such as a group. For example, AddMember adds one or more Active Directory domain users or groups to a user role.



---

Example formats:

-AddMember Domain\User

-AddMember User

-AddMember User@Domain

-AddMember Domain\LabGroupAlias

-AddMember LabGroupAlias (an Active Directory security group, not an email alias)

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-AddScope<ClientObject[]>**

Adds one or more VMM objects to the scope of objects that members of this user role can manage.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Description<String>**

States a description for the specified object.

|                        |       |
|------------------------|-------|
| Aliases                | none  |
| Required?              | false |
| Position?              | named |
| Default Value          | none  |
| Accept Pipeline Input? | false |

---

|                             |       |
|-----------------------------|-------|
| Accept Wildcard Characters? | false |
|-----------------------------|-------|

### **-JobGroup<Guid>**

Specifies an identifier for a series of commands that will run as a set just before the final command that includes the same job group identifier runs.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Name<String>**

Specifies the name of a VMM object.

|               |       |
|---------------|-------|
| Aliases       | none  |
| Required?     | false |
| Position?     | named |
| Default Value | none  |

|                             |       |
|-----------------------------|-------|
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-Permission<SelfServicePermission[]>**

Specifies the actions that members of a Self-Service User role can perform on their virtual machines or services.

Valid values are: AllowLocalAdmin, Author, CanShare, CanReceive, Checkpoint, CheckpointRestoreOnly, Create, CreateFromVHDOrTemplate, PauseAndResume, RemoteConnect, Remove, Save, Shutdown, Start, Stop, Store.

Giving CreateFromVHDOrTemplate permission also gives Create permission. Giving Checkpoint permission also gives CheckpointRestoreOnly permission.

Example format: -Permission Create,PauseAndResume,Stop

The acceptable values for this parameter are:

|                         |  |
|-------------------------|--|
| Create                  |  |
| PauseAndResume          |  |
| Start                   |  |
| Stop                    |  |
| AllowLocalAdmin         |  |
| RemoteConnect           |  |
| Remove                  |  |
| Shutdown                |  |
| Checkpoint              |  |
| Store                   |  |
| Save                    |  |
| Author                  |  |
| CanShare                |  |
| CanReceive              |  |
| CreateFromVHDOrTemplate |  |
| CheckpointRestoreOnly   |  |
| AuthorVMNetwork         |  |

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RemoveLibraryStoreSharePath**

Clears the user role data path for a self-service user.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RemoveMember<String[]>**

Removes a member from a VMM object that has the concept of membership, such as a group. For example, RemoveMember removes one or more Active Directory domain users or groups from a user role.

---

Example formats:

-RemoveMember Domain\User

-RemoveMember User

-RemoveMember User@Domain

-RemoveMember Domain\LabGroupAlias

-RemoveMember LabGroupAlias (an Active Directory security group, not an email alias)

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RemoveScope<ClientObject[]>**

Removes one or more VMM objects from the scope of objects that members of this user role can manage.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RemoveVMNetworkMaximum**

|                        |       |
|------------------------|-------|
| Aliases                | none  |
| Required?              | false |
| Position?              | named |
| Default Value          | none  |
| Accept Pipeline Input? | false |

---

|                             |       |
|-----------------------------|-------|
| Accept Wildcard Characters? | false |
|-----------------------------|-------|

### **-RemoveVMNetworkMaximumPerUser**

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-ShowPROTips<Boolean>**

Indicates whether to show PRO tips. This parameter only applies to Self-Service User roles.

|                        |       |
|------------------------|-------|
| Aliases                | none  |
| Required?              | false |
| Position?              | named |
| Default Value          | none  |
| Accept Pipeline Input? | false |

|                             |       |
|-----------------------------|-------|
| Accept Wildcard Characters? | false |
|-----------------------------|-------|

### **-UserRole<UserRole>**

Specifies a user role object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-UserRoleDataPath<String>**

Specifies the path to a library share that members of a Self-Service User role can use to upload their data.

Example format: "\\LibraryServerName\LibraryShareName"

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | false |
| Position? | named |

---

|                             |       |
|-----------------------------|-------|
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMNetworkMaximum<UInt16>**

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMNetworkMaximumPerUser<UInt16>**

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

### **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **UserRole**



---

## Examples

### 1: Add the specified users to the VMM Administrator user role.

The first command gets the user role object named Administrator and stores the object in the \$UserRole variable.

The second command adds User1 and User2, both members of the Contoso.com domain, to the Administrator user role.

```
PS C:\> $UserRole = Get-SCUserRole -Name "Administrator"
```

```
PS C:\> Set-SCUserRole -UserRole $UserRole -AddMember Contoso\User1,Contoso\User2
```

### 2: Add the specified users to the Administrator role in a single command.

This command gets all user role objects from VMMServer01, selects the user role objects whose profile is Administrator, and then adds User3 to the Administrator user role.

```
PS C:\> Get-SCUserRole -VMMServer "VMMServer01.Contoso.com" | where { $_.Profile -eq "Administrator" } | Set-SCUserRole -AddMember Contoso\User3
```

### 3: Modify an existing self-service user role by adding a cloud to its scope.

The first command gets the cloud object named Cloud02 and stores the object in the \$Cloud variable.

The second command gets the user role object named ContosoSelfServiceUsers and stores the object in the \$UserRole profile.

The last command modifies the scope of the user role stored in \$UserRole (ContosoSelfServiceUsers) by adding the cloud stored in \$Cloud to its scope.

```
PS C:\> $Cloud = Get-SCCloud -Name "Cloud02"
```

```
PS C:\> $UserRole = Get-SCUserRole -Name "ContosoSelfServiceUsers"
```

```
PS C:\> Set-SCUserRole -UserRole $UserRole -AddScope $Cloud
```

### 4: Remove the specified user from the Administrator user role.

The first command gets the user role object named Administrator and stores the object in the \$UserRole variable.

The second command removes User01, who is a member of the Contoso.com domain, from the Administrator user role.

```
PS C:\> $UserRole = Get-SCUserRole -Name "Administrator"
```

```
PS C:\> Set-SCUserRole -UserRole $UserRole -RemoveMember Contoso\User1
```

---

## 5: Add a cloud to the scope of a self-service user role.

The first command gets the cloud object named Cloud03 and stores the object in the \$Cloud variable. The second command gets the user role object named ContosoSelfServiceUsers and then passes the user role object to the Set-SCUserRole cmdlet. The Set-SCUserRole cmdlet adds the cloud stored in \$Cloud to the user role.

```
PS C:\> $Cloud = Get-SCCloud -Name "Cloud03"
PS C:\> Get-SCUserRole -Name "ContosoSelfServiceUsers" | Set-SCUserRole -AddScope $Cloud
```

## 6. Modify what actions members of a self-service user role can take on their virtual machines.

The first command gets the user role object on VMMServer01 named ContosoSelfServiceUsers and stores the object in the \$UserRole variable.

The second command modifies the permissions for members of the user role stored in \$UserRole (ContosoSelfServiceUsers) to allow Creation, PauseAndResume, Stop, AllowLocalAdmin and Store permissions.

To list all available permissions that you can specify for self-service users, type:

```
PS C:\> [enum]::GetValues([Microsoft.VirtualManager.Remoting.SelfServicePermission])
```

You can specify the following permissions with the -Permission parameter:

| PERMISSION              | ALLOWED ACTIONS   |
|-------------------------|---|
| -----                   | -----   |
| Create                  | Create virtual machines and services from VHDs or Templates |
| PauseAndResume          | Pause and resume virtual machines and services              |
| Start                   | Start virtual machines and services                         |
| Stop                    | Stop virtual machines and services                          |
| AllowLocalAdmin         | Act as local Administrator on virtual machines              |
| RemoteConnect           | Access virtual machines remotely                            |
| Remove                  | Remove virtual machines and services                        |
| Shutdown                | Shut down virtual machines                                  |
| Checkpoint              | Create and manage virtual machine checkpoints               |
| Store                   | Store virtual machines in the library                       |
| Save                    | Save virtual machines and services                          |
| Author                  | Author virtual machine and service templates                |
| CanShare                | Share resources with other self-service users               |
| CanReceive              | Receive resources from other self-service users             |
| CreateFromVHDorTemplate | Create virtual machines and services from VHDs or Templates |
| CheckpointRestoreOnly   | Restore to but cannot create virtual machine                |

---

checkpoints

```
PS C:\> $UserRole = Get-SCUserRole -VMMServer "VMMServer01.Contoso.com" -Name  
"ContosoSelfServiceUsers"
```

```
PS C:\> Set-SCUserRole -UserRole $UserRole -Permission  
"Create,PauseAndResume,Stop,AllowLocalAdmin,Store"
```

## Related topics

[Get-SCUserRole](#)

[Grant-SCResource](#)

[New-SCUserRole](#)

[Remove-SCUserRole](#)

[Revoke-SCResource](#)

---

# Set-SCUserRoleQuota

---

## Set-SCUserRoleQuota

Modifies the settings for a user role quota.

### Syntax

Parameter Set: JobGroup

```
Set-SCUserRoleQuota -Cloud <Cloud> -JobGroup <Guid> [-CPUCount <Int32> ] [-CustomQuotaCount <Int32> ] [-JobVariable <String> ] [-MemoryMB <Int32> ] [-PROTipID <Guid> ] [-QuotaPerUser] [-RunAsynchronously] [-StorageGB <Int32> ] [-UseCPUCountMaximum] [-UseCustomQuotaCountMaximum] [-UseMaximumQuota] [-UseMemoryMBMaximum] [-UseStorageGBMaximum] [-UseVMCountMaximum] [-VMCount <Int32> ] [-VMMServer <ServerConnection> ] [ <CommonParameters> ]
```

Parameter Set: UseDefault

```
Set-SCUserRoleQuota -UseMaximumQuota [-JobVariable <String> ] [-PROTipID <Guid> ] [-QuotaPerUser] [-RunAsynchronously] [-UserRoleQuota <UserRoleQuota> ] [-VMMServer <ServerConnection> ] [ <CommonParameters> ]
```

Parameter Set: Values

```
Set-SCUserRoleQuota -UserRoleQuota <UserRoleQuota> [-CPUCount <Int32> ] [-CustomQuotaCount <Int32> ] [-JobVariable <String> ] [-MemoryMB <Int32> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-StorageGB <Int32> ] [-UseCPUCountMaximum] [-UseCustomQuotaCountMaximum] [-UseMemoryMBMaximum] [-UseStorageGBMaximum] [-UseVMCountMaximum] [-VMCount <Int32> ] [-VMMServer <ServerConnection> ] [ <CommonParameters> ]
```

### Detailed Description

The Set-SCUserRoleQuota modifies the settings for a Virtual Machine Manager (VMM) user role quota.

### Parameters

#### **-Cloud<Cloud>**

Specifies a private cloud object.

|           |      |
|-----------|------|
| Aliases   | none |
| Required? | true |

---

|                             |       |
|-----------------------------|-------|
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-CPUCount<Int32>**

Specifies the number of virtual CPUs for a user role quota or cloud capacity.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-CustomQuotaCount<Int32>**

Specifies the number of custom quota points for a user role quota or cloud capacity.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobGroup<Guid>**

Specifies an identifier for a series of commands that will run as a set just before the final command that includes the same job group identifier runs.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

|                             |       |
|-----------------------------|-------|
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-MemoryMB<Int32>**

Specifies the amount of memory in megabytes (MB) for a user role quota or cloud capacity.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-QuotaPerUser**

Indicates whether the cmdlet sets or retrieves user level quotas or member level quotas. Specifying \$True indicates member level quotas. Specifying \$False indicates role level quotas. If the parameter is not used, both quotas are set or returned.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-StorageGB<Int32>**

Specifies the amount of storage in gigabytes (GB) for a user role quota or cloud capacity. This storage amount does not include library storage.

---

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-UseCPUCountMaximum**

Indicates that the maximum number of virtual CPUs is allowed for a user role or cloud capacity. When this parameter is used, no quota is enforced for the virtual CPU dimension.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-UseCustomQuotaCountMaximum**

Indicates that the maximum number of custom quota points is allowed for a user role or cloud capacity. When this parameter is used, no quota is enforced for the custom quota dimension.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |



---

## **-UseMaximumQuota**

Indicates that all quota dimensions are set to maximum. When this parameter is used, no quotas are enforced.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-UseMemoryMBMaximum**

Indicates that the maximum amount of memory, in megabytes (MB), is allowed for a user role or cloud capacity. When this parameter is used, no quota is enforced for the memory dimension.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-UserRoleQuota<UserRoleQuota>**

Specifies a user role quota object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

---

## **-UseStorageGBMaximum**

Indicates that the maximum amount of storage, in gigabytes (GB), is allowed for a user role or cloud capacity. When this parameter is used, no quota is enforced for the storage dimension.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-UseVMCountMaximum**

Indicates that the maximum number of virtual machines is allowed for a user role or cloud capacity. When this parameter is used, no quota is enforced for the virtual machine dimension.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VMCount<Int32>**

Specifies the number of virtual machines for a user role quota or cloud capacity.

|                        |       |
|------------------------|-------|
| Aliases                | none  |
| Required?              | false |
| Position?              | named |
| Default Value          | none  |
| Accept Pipeline Input? | false |

|                             |       |
|-----------------------------|-------|
| Accept Wildcard Characters? | false |
|-----------------------------|-------|

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **UserRoleQuota**

## **Examples**

### **1: Increase virtual machine count quota of a user role for a cloud.**

The first command gets the cloud object named Cloud01 and stores the object in the \$Cloud variable.

The second command gets the user role object named ContosoSelfServiceUsers and stores the object in the \$Role variable.

The third command gets the user role quota for the cloud stored in \$Cloud (Cloud01) and user role stored in \$Role (ContosoSelfServiceUsers). The QuotaPerUser parameter set to false indicates that the quota for the user role will be returned.

The last command determines whether the virtual machine quota is less than 20. If it is, then it sets the quota to 20.

```
PS C:\> $Cloud = Get-SCCloud -Name "Cloud01"
```

```
PS C:\> $Role = Get-SCUserRole -Name "ContosoSelfServiceUsers"
```

```
PS C:\> $Quota = Get-SCUserRoleQuota -Cloud $Cloud -UserRole $Role -QuotaPerUser $False
```

---

```
PS C:\> Write-Output $Quota.VMCount
```

```
PS C:\> if ($Quota.VMCount -lt 20) {Set-SCUserRoleQuota -UserRoleQuota $quota -VMCount 20}
```

## Related topics

[Get-SCUserRoleQuota](#)

---

# Set-SCVirtualCOMPort

---

## Set-SCVirtualCOMPort

Changes properties of a virtual COM port associated with a virtual machine, virtual machine template, or hardware profile.

### Syntax

Parameter Set: NamedPipe

```
Set-SCVirtualCOMPort [-VirtualCOMPort] <VirtualCOMPort> -NamedPipe <String> [-JobGroup <Guid> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

Parameter Set: NewParentJobGroupHostPort

```
Set-SCVirtualCOMPort -GuestPort <Byte> -JobGroup <Guid> -VMHostCOMPort <Byte> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [-WaitForModem <Boolean> ] [ <CommonParameters>]
```

Parameter Set: NewParentJobGroupNamedPipe

```
Set-SCVirtualCOMPort -GuestPort <Byte> -JobGroup <Guid> -NamedPipe <String> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: NewParentJobGroupNoAttach

```
Set-SCVirtualCOMPort -GuestPort <Byte> -JobGroup <Guid> -NoAttach [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: NewParentJobGroupTextFile

```
Set-SCVirtualCOMPort -GuestPort <Byte> -JobGroup <Guid> -TextFile <String> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: NoAttach

```
Set-SCVirtualCOMPort [-VirtualCOMPort] <VirtualCOMPort> -NoAttach [-JobGroup <Guid> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

Parameter Set: TextFile

```
Set-SCVirtualCOMPort [-VirtualCOMPort] <VirtualCOMPort> -TextFile <String> [-JobGroup <Guid> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

Parameter Set: VMHostCOMPort

```
Set-SCVirtualCOMPort [-VirtualCOMPort] <VirtualCOMPort> -VMHostCOMPort <Byte> [-JobGroup <Guid> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-WaitForModem <Boolean> ] [ <CommonParameters>]
```

---

## Detailed Description

The Set-SCVirtualCOMPort cmdlet changes one or more properties of a virtual communications (COM) port associated with a virtual machine, virtual machine template, or hardware profile used in a Virtual Machine Manager (VMM) environment.

### CONNECTING A VIRTUAL COM PORT

-----

Depending on the type of host on which a virtual machine is, or will be, deployed, you can use Set-SCVirtualCOMPort to connect a virtual COM port to a physical COM port on a host server, to a text file, or to a named pipe, or you can use it to disconnect a virtual COM port. Connecting a virtual COM port on a virtual machine to a physical COM port on its host lets the virtual machine use the physical COM port for input and output.

| Type of Host | Available Virtual COM Port Connection Types |
|--------------|---|
|--------------|---|

-----

|         |                               |
|---------|-------------------------------|
| Hyper-V | Connects to a named pipe only |
|---------|-------------------------------|

|            |   |
|------------|---|
| VMware ESX | Connects to a physical COM port, text file, or named pipe |
|------------|---|

|                  |               |
|------------------|---------------|
| Citrix XenServer | Not Supported |
|------------------|---------------|

### THE WAITFORMODEM PARAMETER

-----

You can use the Set-SCVirtualCOMPort cmdlet with the WaitForModem parameter to specify whether a virtual COM port on a virtual machine will connect immediately to a physical COM port on the host when the virtual machine starts.

If WaitForModem is set to TRUE, the virtual machine attempts to connect to the physical COM port on the host only when a program running on the virtual machine sends a modem command to the physical COM port. If the COM port on the host is already connected, the virtual machine cannot connect to it. If the virtual machine successfully connects to the physical COM port, the virtual machine will later release the physical COM port back to the host operating system if the program on the virtual machine that uses the COM port stops using the COM port.

If WaitForModem is set to FALSE, the virtual machine attempts to connect to the physical COM port on the host as soon as the virtual machine starts. If the COM port on the host is already captured, the virtual machine cannot connect to it (same behavior as for TRUE). If the virtual machine successfully connects to the physical COM port, the virtual machine will not release the physical COM port back to the host operating system until the virtual machine is shut down (behavior for FALSE differs from behavior for TRUE).

### PARAMETER SETS THAT USE VIRTUALCOMPORT VERSUS GUESTPORT

-----

The Set-SCVirtualCOMPort cmdlet uses the VirtualCOMPort parameter and the GuestPort parameter as follows:

- VirtualCOMPort <VirtualCOMPort>

Used with four Set-SCVirtualCOMPort parameter sets of to specify a VirtualCOMPort object.

---

- GuestPort <Byte>

Used with four alternate Set-SCVirtualCOMPort parameter sets to specify a virtual COM port by ID (0 or 1).

Review the syntax information for Set-SCVirtualCOMPort to see which parameter sets use the VirtualCOMPort parameter and which use the GuestPort parameter.

## Parameters

### **-GuestPort<Byte>**

Specifies a virtual COM port on a virtual machine by a numerical identifier. Valid values are: 0 or 1.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-JobGroup<Guid>**

Specifies an identifier for a series of commands that will run as a set just before the final command that includes the same job group identifier runs.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-NamedPipe<String>**

Specifies a named pipe to which to connect a virtual COM port. Typical uses include creating a connection between a virtual machine and a debugging program on the host (if the debugger supports the use of named pipes), or creating a virtual null modem cable between two virtual machines.

Example named pipe path: \\.\Contoso\Pipe\PipeName

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-NoAttach**

Specifies that no physical COM port on a host, named pipe, or file will be connected to a virtual COM port; or disconnects a virtual COM port that is already connected to a physical COM port to a named pipe, or to a text file.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |



---

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-TextFile<String>**

Specifies a text file on the host to which to connect a virtual COM port on a virtual machine so that output from the virtual COM port can be sent to that text file. The text file can be on any valid disk drive on the host.

Example format: -TextFile "D:\ComPort.txt"

|                        |       |
|------------------------|-------|
| Aliases                | none  |
| Required?              | true  |
| Position?              | named |
| Default Value          | none  |
| Accept Pipeline Input? | false |

|                             |       |
|-----------------------------|-------|
| Accept Wildcard Characters? | false |
|-----------------------------|-------|

### **-VirtualCOMPort<VirtualCOMPort>**

Specifies a virtual COM port object. VMM supports configuring two COM ports on a virtual machine, template, or hardware profile.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-VMHostCOMPort<Byte>**

Specifies a physical COM port object on a host server to which you can connect a virtual COM port.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|               |       |
|---------------|-------|
| Aliases       | none  |
| Required?     | false |
| Position?     | named |
| Default Value | none  |

|                             |                |
|-----------------------------|----------------|
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-WaitForModem<Boolean>**

Specifies, when set to \$True, that a virtual COM port will wait to connect to a physical COM port on the host, or, when set to \$False, that the virtual COM port will connect immediately to a physical COM port on the host as soon as the virtual machine starts.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **VirtualCOMPort**

## **Notes**

- Requires a VMM virtual COM port object. You can retrieve this object by using the Get-SCVirtualCOMPort cmdlet.

## **Examples**

### **1: Connect a virtual COM port to a named pipe.**

The first command gets the virtual machine object named VM02 and stores the object in the \$VM variable.

The second command gets the virtual COM port named COM1 from VM02 and stores the object in the \$COM1 variable.

---

The last command connects the virtual COM port to the named pipe \\Contoso\Pipe\PipeName.

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM02"
```

```
PS C:\> $COM1 = Get-SCVirtualCOMPort -VM $VM | where {$_.Name -eq "COM1"}
```

```
PS C:\> Set-SCVirtualCOMPort -VirtualCOMPort $COM1 -NamedPipe "\\Contoso\Pipe\PipeName"
```

## 2: Disconnect a virtual COM port.

The first command gets the virtual machine object named VM04 and stores the object in the \$VM variable.

The second command gets the virtual COM port on VM04 named COM1 and stores the port object in the \$COM1 variable.

The last command disconnects the virtual COM port object in \$COM1 by specifying the NoAttach parameter.

NOTE: You can use this command to disconnect a virtual COM port that is currently connected to a physical COM port on a host, to a named pipe, or to a text file.

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM04"
```

```
PS C:\> $COM1 = Get-SCVirtualCOMPort -VM $VM | where {$_.Name -eq "COM1"}
```

```
PS C:\> Set-SCVirtualCOMPort -VirtualCOMPort $COM1 -NoAttach
```

## Related topics

[Get-SCVirtualCOMPort](#)

---

# Set-SCVirtualDiskDrive

---

## Set-SCVirtualDiskDrive

Modifies settings of a virtual disk drive object on a virtual machine or on a virtual machine template in a VMM environment.

### Syntax

Parameter Set: BusChanges

```
Set-SCVirtualDiskDrive -VirtualDiskDrive <VirtualDiskDrive> [-Bus <Byte> ] [-IDE] [-JobGroup <Guid> ] [-JobVariable <String> ] [-LUN <Byte> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-SCSI] [-StorageClassification <StorageClassification> ] [-VolumeType {None | Boot | System | BootAndSystem} ] [ <CommonParameters>]
```

### Detailed Description

The Set-SCVirtualDiskDrive cmdlet modifies settings of a virtual disk drive object on a virtual machine or on a virtual machine template in a Virtual Machine Manager (VMM) environment. You can use this cmdlet to change the Bus type (IDE or SCSI), or to change the Bus and LUN settings to connect a virtual disk drive to a different location on the bus.

### Parameters

#### -Bus<Byte>

Specifies the IDE bus to which to attach a virtual disk drive or virtual DVD drive, or the SCSI bus to which to attach a virtual disk drive.

Example format: -IDE -Bus 1 -LUN 0

Example format: -SCSI -Bus 0 -LUN 1

NOTE: See the examples for a specific cmdlet to determine how that cmdlet uses this parameter.

|                        |       |
|------------------------|-------|
| Aliases                | none  |
| Required?              | false |
| Position?              | named |
| Default Value          | none  |
| Accept Pipeline Input? | false |

---

|                             |       |
|-----------------------------|-------|
| Accept Wildcard Characters? | false |
|-----------------------------|-------|

## **-IDE**

Specifies IDE as the bus type to which to attach a virtual disk drive object or a virtual DVD drive object configured on a virtual machine or on a template.

Example format: -IDE –Bus 0 –LUN 1

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-JobGroup<Guid>**

Specifies an identifier for a series of commands that will run as a set just before the final command that includes the same job group identifier runs.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | false |
| Position? | named |

---

|                             |       |
|-----------------------------|-------|
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-LUN<Byte>**

Specifies the logical unit number (LUN) for a virtual disk drive object or for a virtual DVD drive object on an IDE bus, or for a virtual disk drive object on a SCSI bus.

Example format: -IDE -Bus 1 -LUN 0

Example format: -SCSI -Bus 0 -LUN 1

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-SCSI**

Specifies SCSI as the bus type to which to attach a virtual disk drive object configured on a virtual machine or on a template.

Example format: -SCSI -Bus 0 -LUN 0

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-StorageClassification<StorageClassification>**

Specifies a storage classification object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |



---

## **-VirtualDiskDrive<VirtualDiskDrive>**

Specifies a virtual disk drive object. You can attach either a virtual hard disk (for a virtual machine on any host) or a pass-through disk (for a virtual machine on a Hyper-V host or an ESX host) to a virtual disk drive object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-VolumeType<VolumeType>**

Specifies the volume type for a virtual hard disk. Valid values: Boot, System, BootAndSystem, None. The acceptable values for this parameter are:

|               |  |
|---------------|--|
| None          |  |
| Boot          |  |
| System        |  |
| BootAndSystem |  |

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

---

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **VirtualDiskDrive**

## Notes

- Requires a VMM virtual disk drive object, which can be retrieved by using the Get-SCVirtualDiskDrive cmdlet.

## Examples

### 1: Change the IDE bus and LUN settings for a virtual disk drive on a virtual machine.

The first command gets the virtual machine object VM01 and stores the object in the \$VM variable.

The second command gets the virtual disk drive object on VM01 and stores the object in \$VirtDiskDrive. Using the '@' symbol and parentheses ensures that the command stores the results in an array in case the command returns a single object or a null value.

The last command sets the Bus value to 0 and sets the LUN value to 0 for the virtual disk drive on VM01 if the VM has only one virtual disk drive and is located on the second slot of the first IDE channel.

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM01"
PS C:\> $VirtDiskDrive = @(Get-SCVirtualDiskDrive -VM $VM)
PS C:\> if($VirtDiskDrive.Count -eq 1 -and $VirtDiskDrive[0].Bus -eq 0 -and
$VirtDiskDrive[0].Lun -eq 1){Set-SCVirtualDiskDrive -VirtualDiskDrive $VirtDiskDrive[0] -Bus
0 -LUN 0}
```

### 2: Change the bus type for a virtual disk drive from SCSI to IDE.

The first command gets the virtual machine object named VM02 and stores the object in the \$VM variable.

The second command gets all virtual disk drive objects configured for the virtual machine stored in \$VM and stores the virtual disk drive objects in the \$VirtDiskDrive object array. This example assumes that the virtual disk drive is on a SCSI bus.

The last command sets the Bus type to IDE and connects the second virtual disk drive (specified by \$VirtDiskDrive[1]) to Primary Channel (1) and slot 2 (specified by -Bus 0 and LUN 1).

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM02"
PS C:\> $VirtDiskDrive = Get-SCVirtualDiskDrive -VM $VM
PS C:\> Set-SCVirtualDiskDrive -VirtualDiskDrive $VirtDiskDrive[1] -IDE -Bus 0 -LUN 1
```

---

## Related topics

[Compress-SCVirtualDiskDrive](#)

[Convert-SCVirtualDiskDrive](#)

[Expand-SCVirtualDiskDrive](#)

[Get-SCVirtualDiskDrive](#)

[New-SCVirtualDiskDrive](#)

[Remove-SCVirtualDiskDrive](#)

---

# Set-SCVirtualDVDDrive

---

## Set-SCVirtualDVDDrive

Changes properties of a virtual DVD drive associated with a virtual machine, virtual machine template, or hardware profile used in VMM.

### Syntax

Parameter Set: BusChangesWithVirtualDVDDriveSpecified

```
Set-SCVirtualDVDDrive [-VirtualDVDDrive] <VirtualDVDDrive> [-Bus <Byte> ] [-JobGroup <Guid> ] [-JobVariable <String> ] [-LUN <Byte> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

Parameter Set: AnyHostDriveWithSourceBusAndLunSpecified

```
Set-SCVirtualDVDDrive -AnyVMHostDrive -JobGroup <Guid> -SourceBus <Byte> -SourceLUN <Byte> [-Bus <Byte> ] [-JobVariable <String> ] [-LUN <Byte> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

Parameter Set: AnyHostDriveWithVirtualDVDDriveSpecified

```
Set-SCVirtualDVDDrive [-VirtualDVDDrive] <VirtualDVDDrive> -AnyVMHostDrive [-Bus <Byte> ] [-JobGroup <Guid> ] [-JobVariable <String> ] [-LUN <Byte> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

Parameter Set: BusChangesWithSourceBusAndLunSpecified

```
Set-SCVirtualDVDDrive -JobGroup <Guid> -SourceBus <Byte> -SourceLUN <Byte> [-Bus <Byte> ] [-JobVariable <String> ] [-LUN <Byte> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

Parameter Set: HostDriveWithSourceBusAndLunSpecified

```
Set-SCVirtualDVDDrive -JobGroup <Guid> -SourceBus <Byte> -SourceLUN <Byte> -VMHostDrive <String> [-Bus <Byte> ] [-JobVariable <String> ] [-LUN <Byte> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

Parameter Set: HostDriveWithVirtualDVDDriveSpecified

```
Set-SCVirtualDVDDrive [-VirtualDVDDrive] <VirtualDVDDrive> -VMHostDrive <String> [-Bus <Byte> ] [-JobGroup <Guid> ] [-JobVariable <String> ] [-LUN <Byte> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

Parameter Set: ISOWithSourceBusAndLunSpecified

```
Set-SCVirtualDVDDrive -ISO <ISO> -JobGroup <Guid> -SourceBus <Byte> -SourceLUN <Byte> [-Bus <Byte> ] [-JobVariable <String> ] [-Link] [-LUN <Byte> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

Parameter Set: ISOWithVirtualDVDDriveSpecified

```
Set-SCVirtualDVDDrive [-VirtualDVDDrive] <VirtualDVDDrive> -ISO <ISO> [-Bus <Byte> ] [-JobGroup <Guid> ] [-JobVariable <String> ] [-Link] [-LUN <Byte> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

---

Parameter Set: NoMediaWithSourceBusAndLunSpecified

```
Set-SCVirtualDVDDrive -JobGroup <Guid> -NoMedia -SourceBus <Byte> -SourceLUN <Byte> [-Bus <Byte> ] [-JobVariable <String> ] [-LUN <Byte> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

Parameter Set: NoMediaWithVirtualDVDDriveSpecified

```
Set-SCVirtualDVDDrive [-VirtualDVDDrive] <VirtualDVDDrive> -NoMedia [-Bus <Byte> ] [-JobGroup <Guid> ] [-JobVariable <String> ] [-LUN <Byte> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

## Detailed Description

The Set-SCVirtualDVDDrive cmdlet changes one or more properties of a virtual DVD drive object associated with a virtual machine, virtual machine template, or hardware profile used in a Virtual Machine Manager (VMM) environment.

You can use this cmdlet to connect a virtual DVD drive to a physical DVD drive on a virtual machine host server, to a different location on the IDE bus, or to an ISO image, or you can use it to disconnect the virtual DVD drive.

Most settings that you can configure for a virtual DVD drive on a virtual machine are the same regardless of whether the virtualization platform of the host is Hyper-V, VMware, or Citrix XenServer. All of these virtualization platforms support the following:

- Connecting a virtual DVD drive to a primary or secondary channel on a host.
- Capturing information from a physical CD or DVD drive on the host without specifying a drive letter.
- Capturing information from an image (ISO) file stored in the VMM library.
- Capturing "no media" (used to disconnect a virtual DVD drive from the host drive or from an ISO file).

The only setting that varies for this cmdlet by virtualization platform is whether an ISO file can be used directly from the VMM library:

- Hyper-V host. If you configure a connection to an ISO file in the VMM library, you can choose to use the ISO directly from the library instead of copying it to the host.
- VMware ESX host. If you configure a connection to an ISO file in the VMM library, you cannot use the ISO directly from the library but must instead accept the default, which copies the ISO file to the host.
- Citrix XenServer host. If you configure a connection to an ISO file in the VMM library, you cannot use the ISO directly from the library but must instead accept the default, which copies the ISO file to the host. The

---

host must have at least one ISO repository available with write access and enough storage space to contain the ISO file.

Note: If the virtual DVD drive is configured on a virtual machine that was created by using the Virtual Machine wizard in the Hyper-V Manager Console rather than in the VMM console, you must specify a drive letter. That drive letter will appear in the Properties for that virtual machine in the VMM console.

## Parameters

### **-AnyVMHostDrive**

Indicates that a virtual DVD or floppy drive on a virtual machine will be connected to any corresponding physical drive on a host. This mapping occurs when you deploy a stored virtual machine on a host, or when you use a template or hardware profile to create and deploy a virtual machine on a host.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Bus<Byte>**

Specifies the IDE bus to which to attach a virtual disk drive or virtual DVD drive, or the SCSI bus to which to attach a virtual disk drive.

Example format: -IDE -Bus 1 -LUN 0

Example format: -SCSI -Bus 0 -LUN 1

NOTE: See the examples for a specific cmdlet to determine how that cmdlet uses this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-ISO<ISO>**

Specifies an ISO object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-JobGroup<Guid>**

Specifies an identifier for a series of commands that will run as a set just before the final command that includes the same job group identifier runs.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-Link**

Indicates that a resource should be linked to instead of copied.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-LUN<Byte>**

Specifies the logical unit number (LUN) for a virtual disk drive object or for a virtual DVD drive object on an IDE bus, or for a virtual disk drive object on a SCSI bus.

Example format: -IDE -Bus 1 -LUN 0

Example format: -SCSI -Bus 0 -LUN 1

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-NoMedia**

Disconnects a virtual DVD drive from the host drive or ISO to which it was connected, or disconnects a virtual floppy drive from the host drive or virtual floppy disk to which it was connected.

|                        |       |
|------------------------|-------|
| Aliases                | none  |
| Required?              | true  |
| Position?              | named |
| Default Value          | none  |
| Accept Pipeline Input? | false |



---

|                             |       |
|-----------------------------|-------|
| Accept Wildcard Characters? | false |
|-----------------------------|-------|

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-SourceBus<Byte>**

Specifies the source IDE bus for the drive.

|                        |       |
|------------------------|-------|
| Aliases                | none  |
| Required?              | true  |
| Position?              | named |
| Default Value          | none  |
| Accept Pipeline Input? | false |

|                             |       |
|-----------------------------|-------|
| Accept Wildcard Characters? | false |
|-----------------------------|-------|

### **-SourceLUN<Byte>**

Specifies the source logical unit number (LUN) for a virtual DVD drive object on an IDE bus.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VirtualDVDDrive<VirtualDVDDrive>**

Specifies a virtual DVD drive object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-VMHostDrive<String>**

Specifies a drive on a virtual machine host.

Example formats:

Hyper-V host hard drive: "C:"

Hyper-V host floppy drive: "A:"

VMware ESX host hard drive: "/dev/tools"

VMware ESX host floppy drive: "/dev/sda"

Citrix XenServer host hard drive: "Local storage[99b6212f-b63d-c676-25f9-d6c460992de7]"

Citrix XenServer host floppy drive: Not Supported

---

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **VirtualDVDDrive**

## Notes

- Requires a VMM virtual DVD drive object, which can be retrieved by using the Get-SCVirtualDVDDrive cmdlet.

## Examples

### 1: Connect a virtual DVD drive to a physical DVD drive.

The first command gets the virtual machine object named VM01 and stores the object in the \$VM variable.

The second command gets the virtual DVD drive object that is located on Secondary Channel 0 (specified by -Bus 1 and -LUN 0) on the IDE bus on VM01 and stores the object in the \$DVDDrive variable.

The last command connects the virtual DVD drive object stored in \$DVDDrive to a physical drive on the host (the D: drive). It also deletes any ISO file that the virtual DVD drive used earlier if no other virtual machine currently uses that ISO file.

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM01"
```

```
PS C:\> $DVDDrive = Get-SCVirtualDVDDrive -VM $VM | where { $_.Bus -eq 1 -and $_.LUN -eq 0 }
```

```
PS C:\> Set-SCVirtualDVDDrive -VirtualDVDDrive $DVDDrive -VMHostDrive "E:"
```

---

## 2: Connect a virtual DVD drive to a different location on the IDE bus.

The command gets the virtual machine object named VM02 and stores the object in the \$VM variable. The second command gets the virtual DVD drive object that is located on Secondary Channel 0 (specified by -Bus 1 and -LUN 0) on the IDE bus on VM02 and then stores the virtual DVD drive object in \$DVDDrive.

The last command connects the virtual DVD drive object stored in \$DVDDrive to a different position on the IDE bus by setting the logical unit number (LUN) to 1.

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM02"
PS C:\> $DVDDrive = Get-SCVirtualDVDDrive -VM $VM | where { $_.Bus -eq 1 -and $_.LUN -eq 0 }
PS C:\> Set-SCVirtualDVDDrive -VirtualDVDDrive $DVDDrive -Bus 1 -LUN 1
```

## 3: Disconnect a virtual DVD drive.

The first command gets the virtual machine object named VM03 and stores the object in the \$VM variable.

The second command gets the virtual DVD drive object that is located on Secondary Channel 0 (specified by -Bus 1 and -LUN 0) on the IDE bus on VM03 and then stores the virtual DVD drive object in \$DVDDrive.

The last command uses the NoMedia parameter to disconnect the virtual DVD drive object stored in \$DVDDrive from any host drive or ISO to which it is connected. It also deletes any ISO file that the virtual DVD drive used earlier if no other virtual machine currently uses that ISO file.

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM03"
PS C:\> $DVDDrive = Get-SCVirtualDVDDrive -VM $VM | where { $_.Bus -eq 1 -and $_.LUN -eq 0 }
PS C:\> Set-SCVirtualDVDDrive -VirtualDVDDrive $DVDDrive -NoMedia
```

## 4. Connect a virtual DVD drive on an existing virtual machine to any available physical DVD drive.

The first command gets the virtual machine object named VM04 and stores the object in the \$VM variable.

The last command gets the virtual DVD drive object that is located on the first slot of the Secondary Channel (specified by -Bus 1 and -LUN 0) on the IDE bus on VM04. The command uses the Set-SCVirtualDVDDrive cmdlet with the AnyVMHostDrive parameter to connect the virtual DVD drive to any available physical DVD drive on the host.

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM04"
PS C:\> Set-SCVirtualDVDDrive -AnyVMHostDrive -VirtualDVDDrive (Get-VirtualDVDDrive -VM $VM | where { $_.Bus -eq 1 -and $_.Lun -eq 0 })
```

## Related topics

[Get-SCVirtualDVDDrive](#)

---

[New-SCVirtualDVDDrive](#)

[Remove-SCVirtualDVDDrive](#)

---

# Set-SCVirtualFloppyDisk

---

## Set-SCVirtualFloppyDisk

Changes the properties of a virtual floppy disk used in VMM.

### Syntax

Parameter Set: Default

```
Set-SCVirtualFloppyDisk [-VirtualFloppyDisk] <VirtualFloppyDisk> [-Description <String> ] [-Enabled <Boolean> ] [-FamilyName <String> ] [-JobVariable <String> ] [-Name <String> ] [-Owner <String> ] [-PROTipID <Guid> ] [-Release <String> ] [-RunAsynchronously] [-SharePath <String> ] [-UserRole <UserRole> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: EquivalencySet

```
Set-SCVirtualFloppyDisk [-VirtualFloppyDisks] <List<VirtualFloppyDisk>> -FamilyName <String> -Release <String> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Set-SCVirtualFloppyDisk cmdlet changes one or more properties of a virtual floppy disk used in a Virtual Machine Manager (VMM) environment. A virtual floppy disk file used in VMM is either a Windows-based .vfd file or a VMware-based .flp file.

Properties that you can change include:

- Description
- Enabled
- Name
- Owner
- SharePath

You can store a virtual floppy disk file in the VMM library, or you can add the virtual floppy disk to a virtual machine.

### Parameters

#### **-Description<String>**

States a description for the specified object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Enabled<Boolean>**

Enables an object when set to \$True, or disables an object when set to \$False. For example, if you want to upgrade software on a virtual machine template, you can disable the template object in the VMM library to temporarily prevent users from using that object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-FamilyName<String>**

Specifies a family name for a physical resource in the VMM library. This value is used in conjunction with Release, Namespace, and Type to establish equivalency among library resources.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-Name<String>**

Specifies the name of a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-Owner<String>**

Specifies the owner of a VMM object in the form of a valid domain user account.

Example format: -Owner "Contoso\ReneeLo"

Example format: -Owner "ReneeLo@Contoso"

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |



---

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-Release<String>**

Specifies a string that describes the release of a library resource. VMM automatically creates a release value for every resource imported into the library. After the resource has been imported, the string can be customized.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                        |       |
|------------------------|-------|
| Aliases                | none  |
| Required?              | false |
| Position?              | named |
| Default Value          | none  |
| Accept Pipeline Input? | false |

|                             |       |
|-----------------------------|-------|
| Accept Wildcard Characters? | false |
|-----------------------------|-------|

### **-SharePath<String>**

Specifies a path to a valid library share on an existing library server that uses a Universal Naming Convention (UNC) path.

Example format: -SharePath "\\LibServer01\LibShare"

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-UserRole<UserRole>**

Specifies a user role object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VirtualFloppyDisk<VirtualFloppyDisk>**

Specifies a virtual floppy disk object.

|           |      |
|-----------|------|
| Aliases   | none |
| Required? | true |
| Position? | 1    |

|                             |                |
|-----------------------------|----------------|
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-VirtualFloppyDisks<List<VirtualFloppyDisk>>**

Specifies an array of floppy disk objects.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

### **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **VirtualFloppyDisk**

---

## Notes

- Requires a VMM virtual floppy disk object, which can be retrieved by using the Get-SCVirtualFloppyDisk cmdlet.

## Examples

### 1: Change the description of a virtual floppy disk.

The first command gets the floppy disk object named BootFloppy.vfd from the VMM library on VMMServer01 whose file is stored on LibraryServer01, and then stores the object in the \$VFD variable.

The second command changes the description of the virtual floppy disk object stored in \$VFD to "Latest Boot Floppy".

```
PS C:\> $VFD = Get-SCVirtualFloppyDisk -VMMServer "VMMServer01.Contoso.com" | where {
$_Name -eq "BootFloppy.vfd" -and $_.LibraryServer.Name -eq "LibraryServer01.Contoso.com" }
PS C:\> Set-SCVirtualFloppyDisk -VirtualFloppyDisk $VFD -Description "Latest Boot Floppy"
```

### 2: Disable a virtual floppy disk.

The first command gets the virtual floppy disk object named BootFloppy.vfd whose file is stored on LibraryServer01, and then stores the object in the \$VFD variable.

The second command disables the virtual floppy disk object stored in \$VFD.

```
PS C:\> $VFD = Get-SCVirtualFloppyDisk -VMMServer "VMMServer01.Contoso.com" | where {
$_Name -eq "BootFloppy.vfd" -and $_.LibraryServer.Name -eq "LibraryServer01.Contoso.com" }
PS C:\> Set-SCVirtualFloppyDisk -VirtualFloppyDisk $VFD -Enabled $FALSE
```

### 3: Change the name of a VMware-based virtual floppy disk.

The first command gets the floppy disk object named ESXBootFloppy.flp on LibraryServer01 and then stores the object in the \$FLP variable.

The second command changes the name of the virtual floppy disk object stored in \$FLP to ESXBootFloppy-Beta.flp.

NOTE: By default, the name of a virtual floppy disk object in the library is the same name (including the extension) as the name of the actual virtual floppy disk file on the library share. Changing the name of the virtual floppy disk object in the library does not change the name of the actual virtual floppy disk file stored on the library server.

```
PS C:\> $FLP = Get-SCVirtualFloppyDisk -VMMServer "VMMServer01.Contoso.com" | where {
$_Name -eq "ESXBootFloppy.flp" -and $_.LibraryServer.Name -eq "LibraryServer01.Contoso.com"
}
PS C:\> Set-SCVirtualFloppyDisk -VirtualFloppyDisk $FLP -Name "ESXBootFloppy-Beta.flp"
```

---

## 4: Specify an owner for all virtual floppy disks with an "Unknown" owner.

This command gets all virtual floppy disk objects from the VMM library whose owner is "Unknown", and then specifies an owner for each virtual floppy disk object.

```
PS C:\> Get-SCVirtualFloppyDisk -VMMServer "VMMServer01.Contoso.com" | where {$_.Owner -eq "Unknown"} | Set-SCVirtualFloppyDisk -Owner "Contoso\ReneeLo"
```

### Related topics

[Get-SCVirtualFloppyDisk](#)

[Get-SCVirtualFloppyDrive](#)

[Remove-SCVirtualFloppyDisk](#)

[Set-SCVirtualFloppyDrive](#)

---

# Set-SCVirtualFloppyDrive

---

## Set-SCVirtualFloppyDrive

Changes properties of a virtual floppy drive associated with a virtual machine, virtual nmachine template, or hardware profile used in VMM.

### Syntax

Parameter Set: NoMedia

```
Set-SCVirtualFloppyDrive [[-VirtualFloppyDrive] <VirtualFloppyDrive> ] -NoMedia [-JobGroup <Guid> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: VirtualFloppyDisk

```
Set-SCVirtualFloppyDrive [[-VirtualFloppyDrive] <VirtualFloppyDrive> ] -VirtualFloppyDisk <VirtualFloppyDisk> [-JobGroup <Guid> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Set-SCVirtualFloppyDrive cmdlet changes one or more properties of a virtual floppy drive associated with a virtual machine, virtual machine template, or hardware profile used in a Virtual Machine Manager (VMM) environment.

You can use the Set-VirtualFloppyDrive cmdlet to configure the virtual floppy drive to use a physical floppy drive (typically, drive A:) to read physical floppy disks, to read an existing virtual floppy disk, or to disconnect the virtual floppy disk.

### Parameters

#### **-JobGroup<Guid>**

Specifies an identifier for a series of commands that will run as a set just before the final command that includes the same job group identifier runs.

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | false |
| Position? | named |

---

|                             |       |
|-----------------------------|-------|
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-NoMedia**

Disconnects a virtual DVD drive from the host drive or ISO to which it was connected, or disconnects a virtual floppy drive from the host drive or virtual floppy disk to which it was connected.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | false |

|                             |       |
|-----------------------------|-------|
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VirtualFloppyDisk<VirtualFloppyDisk>**

Specifies a virtual floppy disk object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VirtualFloppyDrive<VirtualFloppyDrive>**

Specifies a virtual floppy drive object.

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | false |



|                             |                |
|-----------------------------|----------------|
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **VirtualFloppyDrive**

## **Notes**

- Requires a VMM virtual floppy drive object, which can be retrieved by using the Get-SCVirtualFloppyDrive cmdlet.

## **Examples**

### **1: Connect a virtual floppy drive to a virtual floppy disk.**

The first command gets the virtual floppy disk object named BootDisk.vfd from VMMServer01 and stores the object in the \$FloppyDisk variable.

---

The second command gets the virtual machine object named VM01 and stores the object in the \$VM variable.

The third command gets the virtual floppy drive object on VM01 and stores the virtual floppy drive object in the \$FloppyDrive object array (in the event there is more than one virtual floppy drive object, the array will store all of the objects).

The last command connects the virtual floppy disk stored in \$FloppyDisk (BootDisk.vfd) to the first virtual floppy drive on VM01.

```
PS C:\> $FloppyDisk = Get-SCVirtualFloppyDisk -VMMServer "VMMServer01.Contoso.com" | where {$_.Name -eq "BootDisk.vfd"}
```

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM01"
```

```
PS C:\> $FloppyDrive = @(Get-SCVirtualFloppyDrive -VM $VM)
```

```
PS C:\> Set-SCVirtualFloppyDrive -VirtualFloppyDrive $FloppyDrive[0] -VirtualFloppyDisk $FloppyDisk
```

## 2: Disconnect a virtual floppy drive.

The first command gets the virtual machine object named VM02 and stores the object in the \$VM variable.

The second command gets the virtual floppy drive object on VM02 and stores the object in \$FloppyDrive.

The last command disconnects the virtual floppy drive object stored in \$FloppyDrive from any host drive or virtual floppy disk to which it was connected by specifying the NoMedia parameter. This command also deletes any virtual floppy disk that the virtual floppy drive used earlier if no other virtual machine currently uses that virtual floppy disk.

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM02"
```

```
PS C:\> $FloppyDrive = @(Get-SCVirtualFloppyDrive -VM $VM)
```

```
PS C:\> Set-SCVirtualFloppyDrive -VirtualFloppyDrive $FloppyDrive[0] -NoMedia
```

## Related topics

[Get-SCVirtualFloppyDisk](#)

[Get-SCVirtualFloppyDrive](#)

---

# Set-SCVirtualHardDisk

---

## Set-SCVirtualHardDisk

Changes the properties of a virtual hard disk object used in VMM.

### Syntax

Parameter Set: Default

```
Set-SCVirtualHardDisk [-VirtualHardDisk] <VirtualHardDisk> [-Description <String> ] [-Enabled <Boolean> ] [-FamilyName <String> ] [-JobGroup <Guid> ] [-JobVariable <String> ] [-Name <String> ] [-OperatingSystem <OperatingSystem> ] [-Owner <String> ] [-PROTipID <Guid> ] [-Release <String> ] [-RunAsynchronously] [-SharePath <String> ] [-UserRole <UserRole> ] [-VirtualizationPlatform {Unknown | VirtualServer | HyperV | VMWareVC | VMWareESX | XENServer} ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: EquivalencySet

```
Set-SCVirtualHardDisk [-VirtualHardDisks] <List<VirtualHardDisk>> -FamilyName <String> -Release <String> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Set-SCVirtualHardDisk cmdlet changes one or more properties of a virtual hard disk object used in a Virtual Machine Manager (VMM) environment. A virtual hard disk file used in VMM is either a Windows-based .vhd file, a Citrix XenServer-based .vhd file, or a VMware-based .vmdk file.

Properties that you can change include:

- Description
- Enabled
- Name
- Operating System
- Owner
- SharePath

A virtual hard disk file might be stored in the VMM library, or it might be attached to a virtual disk drive on a virtual machine or template.

To change the Bus and LUN settings for a virtual disk drive, use the Set-SCVirtualDiskDrive cmdlet.

---

## Parameters

### **-Description<String>**

States a description for the specified object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Enabled<Boolean>**

Enables an object when set to \$True, or disables an object when set to \$False. For example, if you want to upgrade software on a virtual machine template, you can disable the template object in the VMM library to temporarily prevent users from using that object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-FamilyName<String>**

Specifies a family name for a physical resource in the VMM library. This value is used in conjunction with Release, Namespace, and Type to establish equivalency among library resources.

|               |       |
|---------------|-------|
| Aliases       | none  |
| Required?     | false |
| Position?     | named |
| Default Value | none  |

|                             |       |
|-----------------------------|-------|
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobGroup<Guid>**

Specifies an identifier for a series of commands that will run as a set just before the final command that includes the same job group identifier runs.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Name<String>**

Specifies the name of a VMM object.

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | false |
| Position? | named |

|                             |       |
|-----------------------------|-------|
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-OperatingSystem<OperatingSystem>**

Specifies the type of operating system for a virtual machine. To list the names of all available operating systems in VMM, type: "Get-SCOperatingSystem".

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Owner<String>**

Specifies the owner of a VMM object in the form of a valid domain user account.

Example format: -Owner "Contoso\ReneeLo"

Example format: -Owner "ReneeLo@Contoso"

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

---

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Release<String>**

Specifies a string that describes the release of a library resource. VMM automatically creates a release value for every resource imported into the library. After the resource has been imported, the string can be customized.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-SharePath<String>**

Specifies a path to a valid library share on an existing library server that uses a Universal Naming Convention (UNC) path.

Example format: `-SharePath "\\LibServer01\LibShare"`

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-UserRole<UserRole>**

Specifies a user role object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VirtualHardDisk<VirtualHardDisk>**

Specifies a virtual hard disk object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |



---

## **-VirtualHardDisks<List<VirtualHardDisk>>**

Specifies an array of virtual hard disk objects.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-VirtualizationPlatform<VirtualizationPlatform>**

Specifies the virtualization platform of a virtual machine host managed by VMM. Valid values are: HyperV, VMWareESX, XENServer.

The acceptable values for this parameter are:

|               |  |
|---------------|--|
| Unknown       |  |
| VirtualServer |  |
| HyperV        |  |
| VMWareVC      |  |
| VMWareESX     |  |
| XENServer     |  |

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## -VMMServer<ServerConnection>

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **VirtualHardDisk**

## Notes

- Requires a VMM virtual hard disk object, which can be retrieved by using the Get-SCVirtualHardDisk cmdlet.

## Examples

### 1: Change the description of a virtual hard disk.

The first command retrieves the virtual hard disk object named "WindowsServer2003BootVHD.vhd" from the library on VMMServer01 and then stores the object in the \$VHD variable.

The second command changes the description of the virtual hard disk object stored in \$VHD to "Latest Windows Server 2008 R2 Boot VHD".

```
PS C:\> $VHD = Get-SCVirtualHardDisk -VMMServer "VMMServer01.Contoso.com" | where {
$_.LibraryServer.Name -eq "LibraryServer01.Contoso.com" -and $_.Name -eq
"WindowsServer2003BootVHD.vhd"}
```

```
PS C:\> Set-SCVirtualHardDisk -VirtualHardDisk $VHD -Description "Latest Windows Server 2008
R2 Boot VHD"
```

---

## 2: Enable a VMware-based virtual hard disk in the library.

The first command gets the virtual hard disk object named WindowsServer2008.vmdk and then stores the virtual hard disk object in the \$VMDK variable.

The second command disables the virtual hard disk object stored in \$VMDK.

```
PS C:\> $VMDK = Get-SCVirtualHardDisk -VMMServer "VMMServer01.Contoso.com" | where {
$_ .LibraryServer.Name -eq "LibraryServer01.Contoso.com" -and $_.Name -eq
"WindowsServer2008.vmdk"}
PS C:\> Set-SCVirtualHardDisk -VirtualHardDisk $VMDK -Enabled $FALSE
```

## 3: Specify an owner for all virtual hard disks with an "Unknown" owner.

This command gets all virtual hard disk objects from the VMM library whose owner is "Unknown", and then specifies an owner for each virtual hard disk object.

```
PS C:\> Get-SCVirtualHardDisk -VMMServer "VMMServer01.Contoso.com" | where {$_.Owner -eq
"Unknown"} | Set-SCVirtualHardDisk -Owner "Contoso\ReneeLo"
```

## Related topics

[Compress-SCVirtualDiskDrive](#)

[Convert-SCVirtualDiskDrive](#)

[Expand-SCVirtualDiskDrive](#)

[Get-SCVirtualDiskDrive](#)

[New-SCVirtualDiskDrive](#)

[Remove-SCVirtualDiskDrive](#)

[Set-SCVirtualDiskDrive](#)

[Get-SCVirtualHardDisk](#)

[Move-SCVirtualHardDisk](#)

[Remove-SCVirtualHardDisk](#)

---

# Set-SCVirtualHardDiskConfiguration

---

## Set-SCVirtualHardDiskConfiguration

Modifies the virtual hard disk configuration information contained within a virtual machine configuration.

### Syntax

Parameter Set: Default

```
Set-SCVirtualHardDiskConfiguration -VHDConfiguration <VirtualHardDiskConfiguration> [-DeploymentOption {None | UseFastest | UseTarget | UseNetwork | UseSAN | UseDifferencing | UseExistingVirtualDisk} ] [-DestinationLocation <String> ] [-FileName <String> ] [-JobVariable <String> ] [-ParentVirtualHardDiskPath <String> ] [-PinDestinationLocation <Boolean> ] [-PinFileName <Boolean> ] [-PinSourceLocation <Boolean> ] [-PinStorageDisk <Boolean> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-SourceDisk <StandaloneVirtualHardDisk> ] [-StorageClassification <StorageClassification> ] [-StorageDisk <StorageDisk> ] [ <CommonParameters> ]
```

### Detailed Description

The Set-SCVirtualHardDiskConfiguration modifies the virtual hard disk configuration information that is contained within a virtual machine configuration.

### Parameters

#### **-DeploymentOption<DeploymentOption>**

Specifies the deployment option for a virtual hard disk. Valid values are: None, UseFastest, UseTarget, UseNetwork, UseSAN, UseDifferencing, UseExistingVirtualDisk.

The acceptable values for this parameter are:

|                 |  |
|-----------------|--|
| None            |  |
| UseFastest      |  |
| UseTarget       |  |
| UseNetwork      |  |
| UseSAN          |  |
| UseDifferencing |  |

---

|                        |  |
|------------------------|--|
| UseExistingVirtualDisk |  |
|------------------------|--|

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-DestinationLocation<String>**

Specifies the destination path for a virtual hard disk.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-FileName<String>**

Specifies the file name to use when you rename a virtual hard disk file as you add it to a virtual machine.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-ParentVirtualHardDiskPath<String>**

Specifies the path to the parent virtual hard disk to be used when creating a differencing disk.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-PinDestinationLocation<Boolean>**

Indicates whether the destination location chosen by the user is retained during service deployment configuration.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

### **-PinFileName<Boolean>**

Indicates whether the file name chosen by the user is retained during service deployment configuration.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PinSourceLocation<Boolean>**

Indicates whether the source location chosen by the user is retained during service deployment configuration.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PinStorageDisk<Boolean>**

Indicates whether the storage disk chosen by the user is retained during service deployment configuration.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-SourceDisk<StandaloneVirtualHardDisk>**

Specifies the source virtual hard disk.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |



---

## **-StorageClassification<StorageClassification>**

Specifies a storage classification object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-StorageDisk<StorageDisk>**

Specifies a disk on a Hyper-V or VMware ESX host that a virtual machine on that host can use instead of using a virtual hard disk. This disk is referred to as a pass-through disk (the corresponding VMware term is Raw Device Mapping, or RDM). The host disk is either a local hard disk or a logical unit on a Storage Area Network (SAN). VMM lets the virtual machine bypass the host's file system and access the pass-through disk directly.

TYPE OF HOST    PASS-THROUGH DISK SUPPORT

-----    -----

Hyper-V       Supports pass-through disks

Supports converting a pass-through disk to a VHD

VMware ESX   Supports pass-through disks (RDP), but not disk conversion

Citrix XenServer Does not support pass-through disks

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VHDConfiguration<VirtualHardDiskConfiguration>**

Specifies a virtual hard disk configuration object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **VirtualHardDiskConfiguration**

## Examples

### 1: Set the properties of a virtual hard disk configuration.

The first command gets the service configuration object named Service01 and stores the object in the \$ServiceConfig variable.

The second command gets the computer tier configuration for the service configuration object stored in \$ServiceConfig and stores the object in the \$TierConfig variable.

The third command gets the virtual machine configuration for the computer tier configuration stored in \$TierConfig and stores the object in the \$VMConfig variable.

The fourth command gets the virtual hard disk configuration for the first virtual machine configuration stored in \$VMConfig and stores the object in the \$VHDCfg variable.

The fifth command gets the virtual hard disk object named Win2k8R2BaseDisk.vhd from the library and stores the object in the \$VHD variable.

The last command updates the PinSourceLocation property in the virtual hard disk configuration stored in \$VHDCfg for the source virtual hard disk stored in \$VHD to pin the value of the source virtual hard disk, thereby preventing it from being changed during placement.

```
PS C:\> $ServiceConfig = Get-SCServiceConfiguration -Name "Service01"
```

```
PS C:\> $TierConfig = Get-SCComputerTierConfiguration -ServiceConfiguration $ServiceConfig
```

```
PS C:\> $VMConfig = Get-SCVMConfiguration -ComputerTierConfiguration $TierConfig
```

```
PS C:\> $VHDCfg = Get-SCVirtualHardDiskConfiguration -VMConfiguration $VMConfig[0]
```

```
PS C:\> $VHD = Get-SCVirtualHardDisk -Name "Win2k8R2BaseDisk.vhd"
```

---

```
PS C:\> Set-SCVirtualHardDiskConfiguration -VHDConfiguration $VHDConfig -SourceDisk $VHD -  
PinSourceLocation $True
```

## Related topics

[Get-SCComputerTierConfiguration](#)

[Get-SCServiceConfiguration](#)

[Get-SCVirtualHardDisk](#)

[Get-SCVirtualHardDiskConfiguration](#)

[Get-SCVMConfiguration](#)

---

# Set-SCVirtualizationManager

---

## Set-SCVirtualizationManager

Changes the properties of a VMware vCenter Server that is managed by VMM.

### Syntax

Parameter Set: Default

```
Set-SCVirtualizationManager [-VirtualizationManager] <VirtualizationManager> [-Certificate <ClientCertificate> ] [-Credential <VMMCredential> ] [-EnableSecureMode <Boolean> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-TCPPort <UInt32> ] [ <CommonParameters>]
```

### Detailed Description

The Set-SCVirtualizationManager cmdlet changes one or more properties of a VMware vCenter Server that is managed by Virtual Machine Manager (VMM). A vCenter Server manages VMware ESX hosts and VMware-based virtual machines.

Properties that you can change include settings for the TCP port used to connect to the vCenter Server, credentials used to access the vCenter Server, and updating a vCenter Server security certificate.

If a security certificate for a vCenter Server expires or a self-signed certificate is replaced by a certificate from a third-party Certification Authority (CA), you must update both the vCenter Server and VMM:

- First, replace the current vCenter certificate with the new certificate in vCenter. Refer to the VMware documentation for instructions.
- Next, update the certificate in VMM by importing the new certificate into VMM. See example 3 for this cmdlet.

For more information about including a VMware VirtualCenter Server as a virtualization managers in a Virtual Machine Manager environment, type: "Get-Help Add-SCVirtualizationManager -detailed".

### Parameters

#### **-Certificate<ClientCertificate>**

Specifies a security certificate object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Credential<VMMCredential>**

Specifies a credential object or, for some cmdlets, a Run As account object that contains the user name and password of an account that has permission to perform this action. Or, in the case of Restart-SCJob, has permission to complete a restarted task.

For more information about the PSCredential object, type: "Get-Help Get-Credential".

For more information about Run As accounts, type: "Get-Help New-SCRunAsAccount".

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-EnableSecureMode<Boolean>**

Indicates whether VMM communicates with VMware ESX hosts and Citrix XenServer hosts in secure mode. The default value is \$True.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-TCPPort<UInt32>**

Specifies a numeric value that represents a TCP port.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VirtualizationManager<VirtualizationManager>**

Specifies a virtualization manager object managed by VMM.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **VirtualizationManager**

## **Notes**

- Requires a VMM virtualization manager object, which can be retrieved by using the Get-SCVirtualizationManager cmdlet.

---

## Examples

### 1: Specify new credentials for a virtualization manager.

The first command gets the virtualization manager object named VirtMgrServer01 from the VMM database and stores the object in the \$VirtManager variable.

The second command gets the Run As account named Host Computer Account 04 and stores it in the \$Credential variable.

The last command changes the stored credentials for VirtMgrServer01 that are used when VMM connects to the external service.

```
PS C:\> $VirtManager = Get-SCVirtualizationManager -ComputerName  
"VirtMgrServer01.Contoso.com"
```

```
PS C:\> $Credential = Get-SCRunAsAccount -Name "RunAsAccount04"
```

```
PS C:\> Set-SCVirtualizationManager -VirtualizationManager $VirtManager -Credential  
$Credential
```

### Related topics

[Get-SCVirtualizationManager](#)

[Add-SCVirtualizationManager](#)

[Read-SCVirtualizationManager](#)

[Remove-SCVirtualizationManager](#)



---

# Set-SCVirtualMachine

---

## Set-SCVirtualMachine

Changes properties of a virtual machine managed by VMM.

### Syntax

Parameter Set: SingleVM

```
Set-SCVirtualMachine [-VM] <VM> [-AvailabilitySetNames <List<String>> ] [-BlockDynamicOptimization <Boolean> ] [-BootOrder {Floppy | CD | IdeHardDrive | PxeBoot | Unknown} ] [-CapabilityProfile <CapabilityProfile> ] [-Cloud <Cloud> ] [-ClusterNonPossibleOwner <List<Host>> ] [-ClusterPreferredOwner <List<Host>> ] [-CostCenter <String> ] [-CPUCount <Byte> ] [-CPUExpectedUtilizationPercent <Int32> ] [-CPULimitForMigration <Boolean> ] [-CPULimitFunctionality <Boolean> ] [-CPUMaximumPercent <Int32> ] [-CPUPerVirtualNumaNodeMaximum <Byte> ] [-CPURelativeWeight <Int32> ] [-CPUReserve <Int32> ] [-CPUType <ProcessorType> ] [-Custom1 <String> ] [-Custom10 <String> ] [-Custom2 <String> ] [-Custom3 <String> ] [-Custom4 <String> ] [-Custom5 <String> ] [-Custom6 <String> ] [-Custom7 <String> ] [-Custom8 <String> ] [-Custom9 <String> ] [-DelayStartSeconds <Int32> ] [-Description <String> ] [-DiskIops <Int32> ] [-DRProtectionRequired <Boolean> ] [-DynamicMemoryBufferPercentage <Int32> ] [-DynamicMemoryEnabled <Boolean> ] [-DynamicMemoryMaximumMB <Int32> ] [-DynamicMemoryMinimumMB <UInt16> ] [-EnableBackup <Boolean> ] [-Enabled <Boolean> ] [-EnableDataExchange <Boolean> ] [-EnableHeartbeat <Boolean> ] [-EnableOperatingSystemShutdown <Boolean> ] [-EnableTimeSync <Boolean> ] [-HAVMPriority <UInt32> ] [-HighlyAvailable <Boolean> ] [-InstallVirtualizationGuestServices <Boolean> ] [-JobGroup <Guid> ] [-JobVariable <String> ] [-MemoryMB <Int32> ] [-MemoryPerVirtualNumaNodeMaximumMB <Int32> ] [-MemoryWeight <Int32> ] [-MonitorMaximumCount <Int32> ] [-MonitorMaximumResolution <String> ] [-Name <String> ] [-NetworkUtilizationMbps <Int32> ] [-NumaIsolationRequired <Boolean> ] [-NumLock <Boolean> ] [-OperatingSystem <OperatingSystem> ] [-Owner <String> ] [-PROTipID <Guid> ] [-QuotaPoint <UInt32> ] [-RemoveCapabilityProfile] [-RemoveFromCloud] [-RemoveSelfServiceUserRole <Boolean> ] [-RunAsSystem] [-RunAsUserCredential <PSCredential> ] [-RunAsynchronously] [-StartAction {NeverAutoTurnOnVM | AlwaysAutoTurnOnVM | TurnOnVMIIfRunningWhenVSSStopped} ] [-StopAction {SaveVM | TurnOffVM | ShutdownGuestOS} ] [-Tag <String> ] [-UseHardwareAssistedVirtualization <Boolean> ] [-UserRole <UserRole> ] [-VirtualNumaNodesPerSocketMaximum <Byte> ] [-VirtualVideoAdapterEnabled <Boolean> ] [-VMwareResourcePool <VmwResourcePool> ] [ <CommonParameters> ]
```

### Detailed Description

The Set-SCVirtualMachine cmdlet changes one or more properties of a virtual machine managed by Virtual Machine Manager (VMM). Properties that you can change include the following:

-- Associate a virtual machine with a cloud.

- 
- Name, owner, and description of a virtual machine.
  - BIOS boot order (if deployed on a Hyper-V host).
  - Amount of resources on the host used by a virtual machine. These include:
    - Maximum amount of host CPU resources that a virtual machine can use.
    - Expected use of host CPU by a virtual machine.
    - Amount of host CPU resources used by one virtual machine relative to other virtual machines on the same host.
    - Amount of host memory that a virtual machine can use.
    - Amount of bandwidth on the host's network that a virtual machine can use.
  - Hardware settings for a virtual machine unrelated to host resources. These include:
    - Number of CPUs.
    - Type of CPU.
    - Number of disk input/output operations per second (IOPS).
    - Limiting CPU functionality (for an older operating system, such as Windows NT 4.0).
    - Cost center, tag, and custom settings used to filter virtual machines by criteria.
  - Settings that enable various optional capabilities, including:
    - Enabling or disabling a library object to make it available, or temporarily unavailable, to users.
    - Enabling backing up a virtual machine on a Hyper-V host with Volume Shadow Copy service.
    - Enabling a key/value pair for data exchange between a virtual machine and its Hyper-V host.
    - Enabling shutdown of a virtual machine from the Hyper-V console.
    - Enabling time synchronization between a virtual machine and its Hyper-V host.
    - Enabling the BIOS value for NumLock for a virtual machine on a Hyper-V host.
    - Setting that identifies whether a virtual machine is highly available, that is, a virtual machine to be deployed on a node of a Hyper-V host cluster or a Citrix XenServer host cluster.
    - Setting that determines whether virtualization guest services are installed on a virtual machine deployed on a Hyper-V host.
  - Number of seconds to delay before starting a virtual machine.
  - Setting that identifies the operating system used for a virtual machine.
  - Start and stop actions for a virtual machine.
  - Setting that limits the number of virtual machines self-service users can create.
  - Setting used to switch the role that a self-service user who belongs to multiple roles uses to manage a virtual machine.
  - Setting that assigns a virtual machine on an ESX host to a VMware resource pool.

---

If you want to change the properties of a virtual floppy drive, virtual DVD drive, virtual network adapter, or virtual SCSI adapter associated with a specific virtual machine, use the Set-SCVirtualFloppyDrive, Set-SCVirtualDVDDrive, Set-SCVirtualNetworkAdapter, or Set-SCVirtualScsiAdapter cmdlets, respectively.

## Parameters

### **-AvailabilitySetNames<List<String>>**

Specifies a list of availability set names.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-BlockDynamicOptimization<Boolean>**

Indicates whether dynamic optimization is blocked for a virtual machine.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-BootOrder<BootDevice[]>**

Specifies the order of devices that a virtual machine on a Hyper-V host uses to start up. Valid values are: CD, IDEHardDrive, PXEBoot, Floppy.

Example format: -BootOrder PXEBoot,IDEHardDrive,CD,Floppy

The acceptable values for this parameter are:

|              |  |
|--------------|--|
| Floppy       |  |
| CD           |  |
| IdeHardDrive |  |
| PxeBoot      |  |
| Unknown      |  |

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-CapabilityProfile<CapabilityProfile>**

Specifies a capability profile object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Cloud<Cloud>**

Specifies a private cloud object.

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | false |
| Position? | named |

---

|                             |       |
|-----------------------------|-------|
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-ClusterNonPossibleOwner<List<Host>>**

Specifies a list of names of the non-possible cluster node owners for a specified virtual machine.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-ClusterPreferredOwner<List<Host>>**

Specifies a list of names of the preferred cluster node owners for a particular virtual machine.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-CostCenter<String>**

Specifies the cost center for a virtual machine so that you can collect data about the allocation of virtual machines (or resources allocated to virtual machines) to make use of in your billing system.

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | false |

|                             |       |
|-----------------------------|-------|
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-CPUCount<Byte>**

Specifies the number of CPUs on a virtual machine, on a hardware profile, or on a template. See the examples for a specific cmdlet to determine how that cmdlet uses this parameter.

TYPE OF HOST    NUMBER OF PROCESSORS

-----    -----

Hyper-V        Up to 4 CPUs per VM; varies by guest OS

VMware ESX    Up to 4 CPUs per VM for any supported guest OS

Exception: 1 CPU on a VM running Windows NT 4.0

Citrix XenServer Up to 8 CPUs per VM; varies by guest OS

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-CPUExpectedUtilizationPercent<Int32>**

Specifies the percent of CPU on the host that you expect this virtual machine to use. This value is used only when VMM determines a suitable host for the virtual machine.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-CPULimitForMigration<Boolean>**

Limits, when set to \$True, processor features for the specified virtual machine in order to enable migration to a physical computer that has a different version of the same processor as the source computer. VMM does not support migrating virtual machines between physical computers that have processors from different manufacturers.

|                             |                      |
|-----------------------------|----------------------|
| Aliases                     | LimitCPUForMigration |
| Required?                   | false                |
| Position?                   | named                |
| Default Value               | none                 |
| Accept Pipeline Input?      | false                |
| Accept Wildcard Characters? | false                |

## **-CPULimitFunctionality<Boolean>**

Enables running an older operating system (such as Windows NT 4.0) on a virtual machine deployed on a Hyper-V host or on a VMware ESX host by providing limited CPU functionality for the virtual machine.

|                             |                       |
|-----------------------------|-----------------------|
| Aliases                     | LimitCPUFunctionality |
| Required?                   | false                 |
| Position?                   | named                 |
| Default Value               | none                  |
| Accept Pipeline Input?      | false                 |
| Accept Wildcard Characters? | false                 |

## **-CPUMaximumPercent<Int32>**

Specifies the highest percentage of the total resources of a single CPU on the host that can be used by a specific virtual machine at any given time.

Example: -CPUMaximumPercent 80 (to specify 80 per cent)

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-CPUPerVirtualNumaNodeMaximum<Byte>**

Specifies the maximum number of CPUs allowed for each virtual NUMA node.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-CPURelativeWeight<Int32>**

Specifies the amount of CPU resources on a host that this virtual machine can use relative to other virtual machines on the same host. A virtual machine with a higher setting is allocated more CPU resources than a virtual machine with a lower setting.

TYPE OF HOST    RANGE OF RELATIVE VALUES

-----    -----

Hyper-V        1 to 10000

VMware ESX    2000 = High

1500 = Above Normal

1000 = Normal (default)

750 = Below Normal

500 = Low

1 to 1000000 = Custom

The VMware term for these values is "shares."

Citrix XenServer 1 to 65536, normal is 256

NOTE: See the examples for a specific cmdlet to determine how that cmdlet uses this parameter.



|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-CPUReserve<Int32>**

Specifies the minimum percentage of the resources of a single CPU on the host to allocate to a virtual machine. The percentage of CPU capacity that is available to the virtual machine is never less than this percentage.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-CPUType<ProcessorType>**

Specifies the type of CPU for a virtual machine. To retrieve a list of all CPU types that are available for use in virtual machines in a VMM environment, type: "Get-SCCPUType"

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-Custom1<String>**

Specifies a custom property on a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-Custom10<String>**

Specifies a custom property on a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-Custom2<String>**

Specifies a custom property on a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

### **-Custom3<String>**

Specifies a custom property on a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Custom4<String>**

Specifies a custom property on a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Custom5<String>**

Specifies a custom property on a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-Custom6<String>**

Specifies a custom property on a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-Custom7<String>**

Specifies a custom property on a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-Custom8<String>**

Specifies a custom property on a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-Custom9<String>**

Specifies a custom property on a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-DelayStartSeconds<Int32>**

Specifies the number of seconds to wait after the virtualization service starts before automatically starting a virtual machine. This delay is used to stagger the startup time of multiple virtual machines to help reduce the demand on the physical computer's resources. A typical setting might be 30 to 60 seconds.

TYPE OF HOST      MAXIMUM CONFIGURABLE DELAY

-----      -----

Hyper-V      1000000000 seconds (277777 hours)

VMware ESX      65535 seconds (18 hours)

Citrix XenServer      Does not apply to XenServer virtual machines

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-Description<String>**

States a description for the specified object.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-DiskIops<Int32>**

Specifies the number of disk input/output operations per second (IOPS) on the host that can be used by a specific virtual machine.

Example: -DiskIO 1500 (to specify 1500 IOPS).

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-DRProtectionRequired<Boolean>**

This parameter is reserved for future use.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-DynamicMemoryBufferPercentage<Int32>**

Specifies the percentage of memory above a virtual machine's current memory allocation which the host should try to reserve as a buffer. The default value is 20.

---

Example format: -DynamicMemoryTargetBufferPercentage 20

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-DynamicMemoryEnabled<Boolean>**

Enables, when set to \$True, dynamic memory for virtual machines. You can enable dynamic memory directly on a virtual machine, or on a template or hardware profile that will be used to create virtual machines. The default value is False.

REQUIRED: You can enable dynamic memory for a virtual machine only if that virtual machine is deployed on a host running Windows Server 2008 SP1 or later or if the virtual machine is stored in a library in a stopped state (hardware changes to a stored virtual machine can only be made if the virtual machine does not have snapshots). Enabling dynamic memory on a virtual machine stored in a library will limit placement of that machine to hosts running Windows Server 2008 SP1 or later.

Example format: -DynamicMemoryEnabled \$True

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-DynamicMemoryMaximumMB<Int32>**

Specifies the maximum amount of memory that can be allocated to a virtual machine if dynamic memory is enabled. The default value is 65536.

REQUIRED: You can enable dynamic memory for a virtual machine only if that virtual machine is deployed on a host running Windows Server 2008 R2 SP1 or later or if the virtual machine is stored in a library in a stopped state (hardware changes to a stored virtual machine can only be made if the virtual machine does not have snapshots). Enabling dynamic memory on a virtual machine stored in a library will limit placement of that machine to hosts running Windows Server 2008 R2 SP1 or later.

---

Example format: -DynamicMemoryMaximumMB 1024

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-DynamicMemoryMinimumMB<UInt16>**

Specifies the minimum amount of memory that can be allocated to a virtual machine if dynamic memory is enabled. The default value is 65536.

REQUIRED: You can enable dynamic memory for a virtual machine only if that virtual machine is deployed on a host running Windows Server 2008 R2 SP1 or later or if the virtual machine is stored in a library in a stopped state (hardware changes to a stored virtual machine can only be made if the virtual machine does not have snapshots). Enabling dynamic memory on a virtual machine stored in a library will limit placement of that machine to hosts running Windows Server 2008 R2 SP1 or later.

Example format: -DynamicMemoryMinimumMB 1024

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-EnableBackup<Boolean>**

Enables the use of the Volume Shadow Copy service to back up a virtual machine if the virtual machine is deployed on a Hyper-V host.

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | false |
| Position? | named |



---

|                             |       |
|-----------------------------|-------|
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Enabled<Boolean>**

Enables an object when set to \$True, or disables an object when set to \$False. For example, if you want to upgrade software on a virtual machine template, you can disable the template object in the VMM library to temporarily prevent users from using that object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-EnableDataExchange<Boolean>**

Enables the use of a key/value pair for the exchange of data between a virtual machine and the host operating system if the virtual machine is deployed on a Hyper-V host.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-EnableHeartbeat<Boolean>**

Enables the use of a heartbeat (a signal emitted at regular intervals) to monitor the health of a virtual machine deployed on a Hyper-V host.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-EnableOperatingSystemShutdown<Boolean>**

Enables the shut down of the operating system on a virtual machine managed by VMM from Hyper-V's management interfaces on the host if the virtual machine is deployed on a Hyper-V host.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-EnableTimeSync<Boolean>**

Enables synchronizing the system time of a virtual machine with the system time of the operating system running on the host if the virtual machine is deployed on a Hyper-V host.

|                             |                           |
|-----------------------------|---------------------------|
| Aliases                     | EnableTimeSynchronization |
| Required?                   | false                     |
| Position?                   | named                     |
| Default Value               | none                      |
| Accept Pipeline Input?      | false                     |
| Accept Wildcard Characters? | false                     |

---

## **-HAVMPriority<UInt32>**

Specifies the virtual machine priority. Virtual machines are started in priority order up to the limits of a host cluster node. Valid input for this parameter is a numerical representation of the priority:

3000 = High

2000 = Medium

1000 = Low

0 - Virtual machine does not restart automatically.

Example format: -HAVMPriority 2000

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-HighlyAvailable<Boolean>**

Specifies that a virtual machine will be placed on a Hyper-V host that is part of a host cluster. Configure this setting on a virtual machine, or on a template or hardware profile that will be used to create virtual machines.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-InstallVirtualizationGuestServices<Boolean>**

Installs virtualization guest services on a Windows-based virtual machine. By default, this parameter is set to \$False and VMM installs the appropriate virtualization guest service automatically. For a virtual machine on a Hyper-V host, the virtualization guest service is called Integration Components (VMGuest.iso). For a virtual machine on a XenServer host, the virtualization guest service is called

---

Citrix Tools for Virtual Machines (xs-tools.iso). Virtual machines on a VMware ESX host do not use a virtualization guest service.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobGroup<Guid>**

Specifies an identifier for a series of commands that will run as a set just before the final command that includes the same job group identifier runs.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |



---

For a host running Windows Server 2008 R2 SP1 or later, 5000 = Normal, 10000 = High, 0 = Low, 1 to 10000 = Custom.

NOTE: See the examples for a specific cmdlet to determine how that cmdlet uses this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-MonitorMaximumCount<Int32>**

Specifies the maximum number of monitors supported by a virtual video adapter.

Example format: -MonitorMaximumCount 3

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-MonitorMaximumResolution<String>**

Specifies, as a string, the value that represents the maximum possible monitor resolution of a virtual video adapter. Valid values are: "1024x768", "1280x1024", "1600x1200", "1920x1200". Default value: "1280x1024"

Example format: -MonitorResolutionMaximum "1600x1200"

|               |       |
|---------------|-------|
| Aliases       | none  |
| Required?     | false |
| Position?     | named |
| Default Value | none  |

---

|                             |       |
|-----------------------------|-------|
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Name<String>**

Specifies the name of a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-NetworkUtilizationMbps<Int32>**

Specifies, in megabits per second (Mbps), the amount of bandwidth on the host's network that can be used by a specific virtual machine.

Example format: -NetworkUtilization 10

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-NumaIsolationRequired<Boolean>**

Indicates whether NUMA isolation is required.

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | false |

|                             |       |
|-----------------------------|-------|
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-NumLock<Boolean>**

Enables the BIOS value for NumLock on a virtual machine (or on a template or hardware profile that is used to create virtual machines) on a Hyper-V host. This parameter does not apply to virtual machines on VMware ESX hosts, or on Citrix XenServer hosts.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-OperatingSystem<OperatingSystem>**

Specifies the type of operating system for a virtual machine. To list the names of all available operating systems in VMM, type: "Get-SCOperatingSystem".

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-Owner<String>**

Specifies the owner of a VMM object in the form of a valid domain user account.

Example format: -Owner "Contoso\ReneeLo"



---

Example format: -Owner "ReneeLo@Contoso"

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-QuotaPoint<UInt32>**

Specifies a quota that limits the number of virtual machines self-service users can deploy.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-RemoveCapabilityProfile**

Removes one or more specified capability profile objects.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RemoveFromCloud**

Removes the association a virtual machine has with a cloud.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RemoveSelfServiceUserRole<Boolean>**

Removes the specified self-service user role from the permission list of the virtual machine.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-RunAsSystem**

Specifies that a virtual machine runs under the local system account.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RunAsUserCredential<PSCredential>**

Specifies the guest account (domain\account) that a virtual machine runs under.

For enhanced security, create a special account with limited permissions:

FILE TYPE    MINIMUM REQUIRED PERMISSIONS FOR GUEST ACCOUNT

-----

.vmc file    Read Data, Write Data, Execute File

.vmc folder List Folder, Write/Create File (required to save VM state)

.vhd file    Read Data, Read Attributes, Read Extended Attributes,  
Write Data

.vnc file    Execute File, Read Data, Read Attributes, Read  
(required if VM connects to a virtual network)

Note: This parameter does not apply to virtual machines on Hyper-V, VMware ESX hosts or Citrix XenServer hosts.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-StartAction<VMStartAction>**

Specifies the behavior of a virtual machine when the virtualization service (Hyper-V, VMware, or XenServer) starts. Valid values are: AlwaysAutoTurnOnVM, NeverAutoTurnOnVM, TurnOnVMIfRunningWhenVSSStopped.

The acceptable values for this parameter are:

|                                 |  |
|---------------------------------|--|
| NeverAutoTurnOnVM               |  |
| AlwaysAutoTurnOnVM              |  |
| TurnOnVMIfRunningWhenVSSStopped |  |

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-StopAction<VMStopAction>**

Specifies the behavior of the virtual machine when the virtualization service (Hyper-V, VMware, or XenServer) stops. Valid values are: SaveVM, TurnOffVM, ShutdownGuestOS.

The acceptable values for this parameter are:

|        |  |
|--------|--|
| SaveVM |  |
|--------|--|

|                 |  |
|-----------------|--|
| TurnOffVM       |  |
| ShutdownGuestOS |  |

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Tag<String>**

Specifies a word or phrase to associate with an object so that you can search for all objects with the specified set of tags. You can search for a subset of tags, or you can search for the full set of tags.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-UseHardwareAssistedVirtualization<Boolean>**

Specifies that hardware-assisted virtualization is used if it is available.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-UserRole<UserRole>**

Specifies a user role object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VirtualNumaNodesPerSocketMaximum<Byte>**

Specifies the maximum number of NUMA nodes allowed for each socket.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VirtualVideoAdapterEnabled<Boolean>**

Enables, when set to \$True, the Microsoft Synthetic 3D Virtual Video Adapter for virtual machines. You can enable the Virtual Video Adapter directly on a virtual machine, or on a template or hardware profile that will be used to create virtual machines.

**REQUIRED:** You can enable the Microsoft Synthetic 3D Virtual Video Adapter for a virtual machine only if that virtual machine is deployed on a host running Windows Server 2008 R2 SP1 (with the Remote Desktop Services role and Remote Desktop Virtual Graphics role service installed) or later or if the virtual machine is stored in a library in a stopped state (hardware changes to a stored virtual machine can only be made if the virtual machine does not have snapshots). Enabling the Microsoft Synthetic 3D Virtual Video Adapter on a virtual machine stored in a library will limit placement of that machine to hosts running Windows Server 2008 R2 SP1 (with the Remote Desktop Services role and Remote Desktop Virtual Graphics role service installed) or later.

---

Example format: -VirtualVideoAdapterEnabled \$TRUE

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VM<VM>**

Specifies a virtual machine object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-VMwareResourcePool<VmwareResourcePool>**

Assigns a virtual machine deployed on a VMware ESX host or a private cloud to a specific VMware resource pool.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **VirtualMachine**

## Notes

- Requires a VMM virtual machine object, which can be retrieved by using the Get-SCVirtualMachine cmdlet.

## Examples

### 1: Specify an amount of memory for an existing virtual machine.

The first command gets the virtual machine object named VM01 and stores the object in the \$VM variable.

The second determines whether the virtual machine stored in \$VM is in a powered off state. If the virtual machine is not in a powered off state, the command uses the Stop-SCVirtualMachine command to power off the virtual machine. For more information about powering off a virtual machine, type: "Get-Help Stop-SCVirtualMachine -detailed".

The last command changes the memory allocated to VM01 to 1024 MB.

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM01"
PS C:\> if($VM.Status -ne "PowerOff"){Stop-SCVirtualMachine -VM $VM}
PS C:\> Set-SCVirtualMachine -VM $VM -MemoryMB 1024
```

### 2: Change the user role used to manage a virtual machine for a user who belongs to multiple self-service user roles.

The first command gets the virtual machine object named VM02 from VMMServer01 and stores the object in the \$VM variable.

The second command gets the user role object named ContosoSelfServiceUsers and stores the object in the \$SSRole variable.

The last command specifies that members of the self-service user role called SSUserRole3 are now granted the permission to manage the VM called VM02.

NOTE: VMM uses the UserRole parameter to set which virtual machines are managed by the members of a specific self-service user role. Typically, you do not need to use the Set-SCVirtualMachine cmdlet with the UserRole parameter to configure this setting. However, if one or more users are members of multiple self-service user roles and you grant them permission to manage multiple virtual machines on



---

the same host, you might encounter a case where you want to switch which user role is authorized to manage a particular virtual machine. This example illustrates that scenario.

```
PS C:\> $VM = Get-SCVirtualMachine -VMMServer "VMMServer01.Contoso.com" -Name "VM02"
PS C:\> $SSRole = Get-SCUserRole -Name "ContosoSelfServiceUsers"
PS C:\> Set-SCVirtualMachine -VM $VM -UserRole $SSRole
```

### 3. Disable time synchronization on a virtual machine used as a domain controller.

The first command stores the current setting for `$ErrorActionPreference` in variable `$EAP`. This variable will be used later to return the setting to its original value.

The second command sets the action preference to `STOP`. This error action preference changes an error from a non-terminating error to a terminating error. The error object is thrown as an exception instead of being written to the output pipe, and the command does not continue to run.

The third command gets the virtual machine object named `VM03` and stores the object in the `$VM` variable.

The fourth command disables the time synchronization setting. Typically, disabling time synchronization is required for virtual machines that act as domain controllers. The command uses the `trap` statement to catch terminating exceptions. If the `Set-SCVirtualMachine` command fails, the string in the `trap` statement is displayed. `Continue` is used in the `trap` statement to continue running instead of exiting. The `Out-Null` cmdlet redirects the output to `$Null` instead of sending it to the console.

The last command sets the value for `$ErrorActionPreference` to the value stored in `$EAP`.

```
PS C:\> $EAP = $ErrorActionPreference
PS C:\> $ErrorActionPreference = "STOP"
PS C:\> $VM = Get-SCVirtualMachine -Name "VM03"
PS C:\> trap{"Fail: Cannot disable Time Synchronization for VM: $VM";continue} Set-
SCVirtualMachine -VM $VM -EnableTimeSynchronization $TRUE | Out-Null
PS C:\> $ErrorActionPreference = $EAP
```

### 4. Set the device boot order for all virtual machines that support this feature.

The first command stores the current setting for `$ErrorActionPreference` in `$EAP`. This variable will be used later to return the setting to its original value.

The second command sets the error action preference to `Stop`. This error action preference changes an error from a non-terminating error to a terminating error. The error object is thrown as an exception instead of being written to the output pipe, and the command does not continue to run.

The third command gets each virtual machine object stores the objects in `$VMs`. Using the `'@'` symbol and parentheses ensures that the command stores the results in an array in case the command returns a single object or a null value.

The fourth command sets the BIOS boot order for each virtual machine to `PXEBoot,IDEHarddrive,CD,Floppy`. The command uses `trap` statement to catch terminating exceptions.

---

If the Set-SCVirtualMachine command fails, the string in the trap statement is displayed. Continue is used in the trap statement to continue running instead of exiting the foreach loop. The Out-Null cmdlet redirects the output to \$Null instead of sending it to the console.

NOTE: The BootOrder parameter is used only for virtual machines on Hyper-V and Citrix XenServer hosts; it is not used for virtual machines on VMware ESX hosts. XenServer hosts do not support floppy disks, and therefore will ignore "Floppy" if listed in the boot order.

The last command sets the value for \$ErrorActionPreference to the value stored in EAP.

For more information about the standard Windows PowerShell foreach loop statement, type: "Get-Help about\_ForEach".

```
PS C:\> $EAP = $ErrorActionPreference
PS C:\> $ErrorActionPreference = "Stop"
PS C:\> $VMs = @(Get-SCVirtualMachine)
PS C:\> foreach($VM in $VMs){trap{"Fail: Cannot set BIOS for VM: $VM";continue} Set-
SCVirtualMachine -VM $VM -BootOrder "PXEBoot","IDEHarddrive","CD","Floppy" | Out-Null}
PS C:\> $ErrorActionPreference = $EAP
```

## 5: Specify an owner for all virtual machines without an owner.

This command gets all virtual machine objects on VMMServer01, selects only those virtual machine objects where no owner is listed, and specifies an owner for each virtual machine.

```
PS C:\> Get-SCVirtualMachine -VMMServer "VMMServer01.Contoso.com" | where {$_.Owner -eq ""}
| Set-SCVirtualMachine -Owner "Contoso\ReneeLo"
```

## 6: Enable Dynamic Memory for an existing virtual machine.

The first command gets the virtual machine object VM06 and stores object in the \$VM variable. To enable Dynamic Memory on a virtual machine, the virtual machine must reside on a host that is running Windows Server 2008 R2 SP1 or later.

The second command determines whether the virtual machine stored in \$VM is in a powered off state. If the virtual machine is not in a powered off state, the command uses the Stop-SCVirtualMachine command to power off the virtual machine. For more information about powering off a virtual machine, type: "Get-Help Stop-SCVirtualMachine -detailed".

The last command enables Dynamic Memory, sets the startup memory to 1024 MB (this is the amount of memory on the host that will be allocated to VM06 upon startup) and the maximum memory to 2048 MB (this is the maximum amount of memory on the host that will be allocated to VM06).

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM06"
PS C:\> if($VM.Status -ne "PowerOff"){Stop-SCVirtualMachine -VM $VM}
PS C:\> Set-SCVirtualMachine -VM $VM -DynamicMemoryEnabled $True -MemoryMB 1024 -
DynamicMemoryMaximumMB 2048
```

---

## Related topics

[Move-SCVirtualMachine](#)

[New-SCVirtualMachine](#)

[Read-SCVirtualMachine](#)

[Remove-SCVirtualMachine](#)

[Repair-SCVirtualMachine](#)

[Resume-SCVirtualMachine](#)

[Save-SCVirtualMachine](#)

[Start-SCVirtualMachine](#)

[Stop-SCVirtualMachine](#)

[Suspend-SCVirtualMachine](#)

---

# Set-SCVirtualNetwork

---

## Set-SCVirtualNetwork

Changes the properties of a virtual network configured on a host managed by VMM.

### Syntax

Parameter Set: Host

```
Set-SCVirtualNetwork [-VirtualNetwork] <VirtualNetwork> [-BoundToVMHost <Boolean> ] [-Description <String> ] [-HostBoundVlanId <UInt16> ] [-JobGroup <Guid> ] [-JobVariable <String> ] [-Name <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

Parameter Set: Cluster

```
Set-SCVirtualNetwork [-ClusterVirtualNetwork] <ClusterVirtualNetwork> [-BoundToVMHost <Boolean> ] [-Description <String> ] [-HostBoundVlanId <UInt16> ] [-JobGroup <Guid> ] [-JobVariable <String> ] [-Name <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

Parameter Set: LogicalSwitch

```
Set-SCVirtualNetwork [-VirtualNetwork] <VirtualNetwork> -LogicalSwitch <LogicalSwitch> [-Description <String> ] [-JobGroup <Guid> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMHostNetworkAdapters <HostNetworkAdapter[]> ] [ <CommonParameters>]
```

### Detailed Description

The Set-SCVirtualNetwork cmdlet changes the properties of a virtual network configured on a host managed by Virtual Machine Manager (VMM).

Virtual network properties that you can change include:

- ANY HOST - For a virtual network configured for virtual machines deployed on any host supported by VMM (a Hyper-V, VMware ESX, or Citrix XenServer host), you can set or modify the name or description.
- HYPER-V HOST ONLY - If the host is a Hyper-V host, you can also configure whether virtual machines are bound to the host (and can thus access the host operating system), and you can specify a numerical identifier for a virtual local area network (VLAN) on the host.

---

## Parameters

### **-BoundToVMHost<Boolean>**

Indicates whether a virtual network is bound to a host. Binding a virtual network to a host enables network communication to the host.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-ClusterVirtualNetwork<ClusterVirtualNetwork>**

Specifies a cluster virtual network object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-Description<String>**

States a description for the specified object.

|                        |       |
|------------------------|-------|
| Aliases                | none  |
| Required?              | false |
| Position?              | named |
| Default Value          | none  |
| Accept Pipeline Input? | false |

---

|                             |       |
|-----------------------------|-------|
| Accept Wildcard Characters? | false |
|-----------------------------|-------|

### **-HostBoundVlanId<UInt16>**

Assigns a VLAN to the virtual network adapter that was created for the host for the specified virtual network.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobGroup<Guid>**

Specifies an identifier for a series of commands that will run as a set just before the final command that includes the same job group identifier runs.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | false |
| Position? | named |

---

|                             |       |
|-----------------------------|-------|
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-LogicalSwitch<LogicalSwitch>**

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-Name<String>**

Specifies the name of a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | false |
| Position? | named |

|                             |       |
|-----------------------------|-------|
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VirtualNetwork<VirtualNetwork>**

Specifies a virtual network object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-VMHostNetworkAdapters<HostNetworkAdapter[]>**

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | false |
| Position? | named |



---

|                             |                |
|-----------------------------|----------------|
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **VirtualNetwork**

## Examples

### 1: Unbind a virtual network from a host.

The first command gets the host object named VMHost01 and stores the object in the \$VMHost variable.

The second command gets the virtual network object named InternalVNet01 from VMHost01 and stores the object in the \$VirtualNetwork variable.

The last command renames the virtual network to "UnboundVNet01" and sets -VMHostBound to FALSE. This unbinds the virtual network from the host, which prevents any virtual machines that are attached to this virtual network from accessing the host through this network.

```
PS C:\> $VMHost = Get-SCVMHost -ComputerName "VMHost01.Contoso.com"
PS C:\> $VirtualNetwork = Get-SCVirtualNetwork -VMHost $VMHost -Name "InternalVNet01"
PS C:\> Set-SCVirtualNetwork -VirtualNetwork $VirtualNetwork -Name "UnboundVNet01" -
BoundToVMHost $False
```

## Related topics

[Add-SCVMHostNetworkAdapter](#)

[Get-SCVirtualNetwork](#)

[Get-SCVirtualNetworkAdapter](#)

[Get-SCVMHostNetworkAdapter](#)

[New-SCVirtualNetwork](#)

[Remove-SCVirtualNetwork](#)

[Set-SCVirtualNetworkAdapter](#)

[Set-SCVMHostNetworkAdapter](#)

---

# Set-SCVirtualNetworkAdapter

---

## Set-SCVirtualNetworkAdapter

Changes properties of a virtual network adapter associated with a virtual machine, a virtual machine template, or a hardware profile used to create virtual machines in VMM.

### Syntax

Parameter Set: IsCheckpoint

```
Set-SCVirtualNetworkAdapter [-VirtualNetworkAdapter] <VirtualNetworkAdapter> ] -Checkpoint  
-VirtualNetwork <String> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously]  
[-VlanEnabled <Boolean> ] [-VlanID <UInt16> ] [-VMMServer <ServerConnection> ] [  
<CommonParameters>]
```

Parameter Set: SlotIdSpecified

```
Set-SCVirtualNetworkAdapter -JobGroup <Guid> -SlotID <Int32> [-EnableMACAddressSpoofing  
<Boolean> ] [-EnableVMNetworkOptimization <Boolean> ] [-IPv4AddressType {Dynamic | Static} ]  
[-IPv6AddressType {Dynamic | Static} ] [-JobVariable <String> ] [-LogicalNetwork  
<LogicalNetwork> ] [-MACAddress <String> ] [-MACAddressType <String> ] [-NetworkLocation  
<String> ] [-NetworkTag <String> ] [-NoConnection] [-NoLogicalNetwork] [-  
NoPortClassification] [-PortClassification <PortClassification> ] [-PROTipID <Guid> ] [-  
RequiredBandwidth <Decimal> ] [-RunAsynchronously] [-VirtualNetwork <String> ] [-VlanEnabled  
<Boolean> ] [-VlanID <UInt16> ] [-VMMServer <ServerConnection> ] [-VMNetwork <VMNetwork> ]  
[-VMNetworkServiceSetting <String> ] [-VMSubnet <VMSubnet> ] [-VMwarePortGroup <String> ] [  
<CommonParameters>]
```

Parameter Set: VirtualNicSpecified

```
Set-SCVirtualNetworkAdapter [-VirtualNetworkAdapter] <VirtualNetworkAdapter> [-  
EnableMACAddressSpoofing <Boolean> ] [-EnableVMNetworkOptimization <Boolean> ] [-IPv4Address  
<String> ] [-IPv4AddressPool <StaticIPAddressPool> ] [-IPv4AddressType {Dynamic | Static} ]  
[-IPv6Address <String> ] [-IPv6AddressPool <StaticIPAddressPool> ] [-IPv6AddressType  
{Dynamic | Static} ] [-JobGroup <Guid> ] [-JobVariable <String> ] [-LogicalNetwork  
<LogicalNetwork> ] [-MACAddress <String> ] [-MACAddressType <String> ] [-NetworkLocation  
<String> ] [-NetworkTag <String> ] [-NoConnection] [-NoLogicalNetwork] [-  
NoPortClassification] [-PortClassification <PortClassification> ] [-PROTipID <Guid> ] [-  
RequiredBandwidth <Decimal> ] [-RunAsynchronously] [-VirtualNetwork <String> ] [-VlanEnabled  
<Boolean> ] [-VlanID <UInt16> ] [-VMMServer <ServerConnection> ] [-VMNetwork <VMNetwork> ]  
[-VMNetworkServiceSetting <String> ] [-VMSubnet <VMSubnet> ] [-VMwarePortGroup <String> ] [  
<CommonParameters>]
```

---

## Detailed Description

The Set-SCVirtualNetworkAdapter cmdlet changes one or more properties of a virtual network adapter associated with a virtual machine, virtual machine template, or hardware profile used to create virtual machines in a Virtual Machine Manager (VMM) environment.

Properties you can change include the following:

- Connect a virtual network adapter to a virtual network.
- Disconnect a virtual network adapter from a virtual network.
- Specify a network location and network tag on a virtual network adapter.
- Specify a MAC address on the virtual network adapter.
- Enable the use of a virtual local area network (VLAN) and specify a VLAN ID (numerical identifier) for that VLAN on the virtual network adapter.

## Parameters

### -Checkpoint

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### -EnableMACAddressSpoofing<Boolean>

Enables, when set to \$True, MAC Address spoofing.

|                             |                             |
|-----------------------------|-----------------------------|
| Aliases                     | MACAddressesSpoofingEnabled |
| Required?                   | false                       |
| Position?                   | named                       |
| Default Value               | none                        |
| Accept Pipeline Input?      | false                       |
| Accept Wildcard Characters? | false                       |

---

## **-EnableVMNetworkOptimization<Boolean>**

Enables, when set to \$True, virtual machine network optimization. This feature improves network performance for virtual machines with network adapters that support virtual machine queue (VMQ) or TCP Chimney Offload. VMQ enables creating a unique network queue for each virtual network adapter. TCP Chimney Offload enables network traffic processing to be offloaded from the networking stack.

|                             |                              |
|-----------------------------|------------------------------|
| Aliases                     | VMNetworkOptimizationEnabled |
| Required?                   | false                        |
| Position?                   | named                        |
| Default Value               | none                         |
| Accept Pipeline Input?      | false                        |
| Accept Wildcard Characters? | false                        |

## **-IPv4Address<String>**

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-IPv4AddressPool<StaticIPAddressPool>**

|                        |       |
|------------------------|-------|
| Aliases                | none  |
| Required?              | false |
| Position?              | named |
| Default Value          | none  |
| Accept Pipeline Input? | false |

---

|                             |       |
|-----------------------------|-------|
| Accept Wildcard Characters? | false |
|-----------------------------|-------|

### **-IPv4AddressType<EthernetAddressType>**

Specifies an IPv4 address type.

The acceptable values for this parameter are:

|         |  |
|---------|--|
| Dynamic |  |
| Static  |  |

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-IPv6Address<String>**

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-IPv6AddressPool<StaticIPAddressPool>**

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | false |

|                             |       |
|-----------------------------|-------|
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-IPv6AddressType<EthernetAddressType>**

Specifies an IPv6 address type.

The acceptable values for this parameter are:

|         |  |
|---------|--|
| Dynamic |  |
| Static  |  |

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobGroup<Guid>**

Specifies an identifier for a series of commands that will run as a set just before the final command that includes the same job group identifier runs.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-LogicalNetwork<LogicalNetwork>**

Specifies a logical network. A logical network is a named grouping of IP subnets and VLANs that is used to organize and simplify network assignments.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-MACAddress<String>**

Specifies the Media Access Control (MAC) address, or a set of MAC addresses, for a physical or virtual network adapter on a computer. Valid values are: one or more MAC addresses.

Example format for a single MAC address:

`-MACAddress "00-15-5D-B4-DC-00"`

Example format for a set of MAC addresses:

`-MACAddress "00-15-5D-B4-DC-00", "00-1A-A0-E3-75-29"`

Example format for a set of MAC addresses:

`$Macs = "00-15-5D-B4-DC-00", "00-1A-A0-E3-75-29"`

`Set-SCPXEserver -MACAddress $Macs`

NOTE: When used with `New-SCPXEserver` or `Set-SCPXEserver`, the `MACAddress` parameter updates the PXE interfaces from which the SCDM PXE Server listens for and responds to PXE requests

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-MACAddressType<String>**

Specifies the type of MAC address to use for a virtual network adapter. Valid values are: Static, Dynamic.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-NetworkLocation<String>**

Specifies the network location for a physical network adapter or for a virtual network adapter, or changes the default network location of a host's physical network adapter.

Example formats:

-NetworkLocation \$NetLoc (\$NetLoc might contain "Corp.Contoso.com")

-OverrideNetworkLocation \$TRUE -NetworkLocation "HostNICNewLocation.Contoso.com"

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |



---

## **-NetworkTag<String>**

Specifies a word or phrase to associate with a virtual network adapter that is configured to connect to a specific internal or external network on the host. The NetworkTag identifies all virtual machines with the same NetworkTag as members of the same network. VMM uses a NetworkTag (if one exists) when it evaluates hosts as possible candidates on which to deploy a virtual machine. If the host does not include virtual machines on the network with the same NetworkTag as the virtual machine to be placed, the host receives zero stars in the placement process.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-NoConnection**

Disconnects a virtual network adapter from a virtual network.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-NoLogicalNetwork**

Indicates that no logical network is associated with this virtual network adapter.

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | false |
| Position? | named |

---

|                             |       |
|-----------------------------|-------|
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-NoPortClassification**

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PortClassification<PortClassification>**

Specifies the port classification of the switch to which the virtual network adapter connects.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | false |
| Position? | named |

---

|                             |       |
|-----------------------------|-------|
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RequiredBandwidth<Decimal>**

Specifies the network bandwidth required by a network adapter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-SlotID<Int32>**

Specifies a numerical ID used to identify a device.

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | true  |
| Position? | named |

|                             |       |
|-----------------------------|-------|
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VirtualNetwork<String>**

Specifies a virtual network object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VirtualNetworkAdapter<VirtualNetworkAdapter>**

Specifies a virtual network adapter object for a virtual machine.

TYPE OF HOST      NUMBER OF VIRTUAL NETWORK ADAPTERS

-----      -----

Hyper-V            Up to 4 emulated adapters per virtual machine.

Up to 8 synthetic adapters per virtual machine.

(Exception: no driver available for an emulated network adapter on a Windows Server 2003 x64 guest.)

VMware ESX      Up to 4 emulated adapters per virtual machine.

Citrix XenServer Up to 7 emulated adapters per virtual machine.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

---

## **-VlanEnabled<Boolean>**

Enables a virtual LAN (VLAN) for use by virtual machines on a Hyper-V or Citrix XenServer host.

Example format for a single VLAN: -VLANEnabled -VLANMode "Access" -VLANID 35

Example format for multiple VLANs: -VLANEnabled -VLANMode "Trunk" -VLANTrunkID  
1,2,100,200,1124

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VlanID<UInt16>**

Assigns a numerical identifier in the range 1-4094 to a virtual network adapter on a virtual machine or to a physical network adapter on a virtual machine host.

Configure a VlanID on a Hyper-V, VMware ESX, or Citrix XenServer host:

- On an externally bound physical network adapter when the Vlan mode is Access.

Configure a VlanID on a virtual network adapter of a virtual machine:

- Bound to a physical network adapter on the host, or
- Bound to an internal virtual network on the host.

Example format: -VlanEnabled

-VlanMode "Access" -VLANID 35

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

---

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-VMNetwork<VMNetwork>**

Specifies a VM network object.

To get a VM network object, use the Get-SCVMNetwork cmdlet.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMNetworkServiceSetting<String>**

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMSubnet<VMSubnet>**

Specifies a VM subnet object.

---

To obtain a VM subnet object, use the Get-SCVMSubnet cmdlet.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VMwarePortGroup<String>**

Specifies the VMware port group.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **VirtualNetworkAdapter**

## **Notes**

- Requires a VMM virtual network adapter object, which can be retrieved by using the Get-SCVirtualNetworkAdapter cmdlet.

---

## Examples

### 1: Connect a virtual network adapter to a virtual network.

The first command gets the virtual machine object named VM01 and stores the object in the \$VM variable.

The second command gets all virtual network adapter objects on VM01, selects the adapter object with the physical (MAC) address of 00:16:D3:CC:00:1B, and then stores the object in the \$Adapter variable.

The last command connects the virtual network adapter stored in \$Adapter to the virtual network named ExternalVirtualNetwork01 on the host that contains VM01.

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM01"
PS C:\> $Adapter = Get-SCVirtualNetworkAdapter -VM $VM | where { $_.PhysicalAddress -eq "00:16:D3:CC:00:1B" }
PS C:\> Set-SCVirtualNetworkAdapter -VirtualNetworkAdapter $Adapter -VirtualNetwork "ExternalVirtualNetwork01"
```

### 2: Specify a static MAC address for a virtual network adapter.

The first command gets the virtual machine object named VM02 and stores the object in the \$VM variable.

The second command gets all virtual network adapter objects on VM02, selects the virtual network adapter with the specified ID, and then stores the object in the \$Adapter variable. This example assumes that this adapter currently has a dynamic MAC address.

The last command specifies that the virtual network adapter stored in \$Adapter use the static MAC address 00:16:D3:CC:00:1C.

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM02"
PS C:\> $Adapter = Get-SCVirtualNetworkAdapter -VM $VM | where { $_.ID -eq "5c0ee80a-731f-41c8-92f0-85a1619f9a1b" }
PS C:\> Set-SCVirtualNetworkAdapter -VirtualNetworkAdapter $Adapter -PhysicalAddressType "Static" -PhysicalAddress "00:16:D3:CC:00:1C"
```

### 3: Specify a static MAC address and assign it to an existing virtual network adapter.

The first command gets the virtual machine object named VM03 and stores the object in the \$VM variable.

The second command gets the virtual network adapter object on VM03 by ID, specifies that the adapter uses a static MAC address type, and assigns it a MACAddress.

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM03"
PS C:\> Set-SCVirtualNetworkAdapter -VirtualNetworkAdapter (Get-VirtualNetworkAdapter -VM $VM | where { $_.ID -eq "95e9cfda-861c-44a3-b2ba-2f796dfe691c" }) -MACAddressType "Static" -MACAddress "00-00-00-00-00-00"
```



---

## 4. Disconnect the specified virtual network adapter from the virtual network.

The first command gets the virtual machine object named VM04 and stores the object in the \$VM variable.

The second command gets all virtual network adapter objects on VM04 and stores the adapter objects in \$Adapters. This example assumes that VM04 has at least 2 virtual network adapters.

The last command uses the NoConnection parameter to disconnect the second virtual network adapter (Adapters[1]) from any virtual network that it is connected to.

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM04"
```

```
PS C:\> $Adapters = Get-SCVirtualNetworkAdapter -VM $VM
```

```
PS C:\> Set-SCVirtualNetworkAdapter -VirtualNetworkAdapter $Adapters[1] -NoConnection
```

## 5: Specify a VMware port group for an existing virtual machine.

The first command gets the virtual machine object named VM05 and stores the object in the \$VM variable.

The second command stores the first [0] virtual network adapter on VM05 in the \$Adapter variable.

The last command sets the virtual network adapter for the adapter stored in \$Adapter to "VM Network", which is the name of the VMware port group that you want this adapter to connect to.

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM05"
```

```
PS C:\> $Adapter = $VM.VirtualNetworkAdapters[0]
```

```
PS C:\> Set-SCVirtualNetworkAdapter -VirtualNetworkAdapter $Adapter -VMwarePortGroup "VM Network"
```

## Related topics

[Add-SCVMHostNetworkAdapter](#)

[Get-SCVirtualNetwork](#)

[Get-SCVirtualNetworkAdapter](#)

[New-SCVirtualNetwork](#)

[New-SCVirtualNetworkAdapter](#)

[Remove-SCVirtualNetworkAdapter](#)

[Set-SCVirtualNetwork](#)

[Set-SCVMHostNetworkAdapter](#)

---

# Set-SCVirtualNetworkAdapterConfiguration

---

## Set-SCVirtualNetworkAdapterConfiguration

Modifies the virtual network adapter configuration contained within a virtual machine configuration.

### Syntax

Parameter Set: Default

```
Set-SCVirtualNetworkAdapterConfiguration -VirtualNetworkAdapterConfiguration  
<VirtualNetworkAdapterConfiguration> [-IPv4Address <String> ] [-IPv4AddressPool  
<StaticIPAddressPool> ] [-IPv4PAAAddress <String> ] [-IPv4PAAAddressPool <StaticIPAddressPool>  
] [-IPv6Address <String> ] [-IPv6AddressPool <StaticIPAddressPool> ] [-IPv6PAAAddress  
<String> ] [-IPv6PAAAddressPool <StaticIPAddressPool> ] [-JobVariable <String> ] [-MACAddress  
<String> ] [-MACAddressPool <MACAddressPool> ] [-PinIPv4AddressPool <Boolean> ] [-  
PinIPv4PAAAddressPool <Boolean> ] [-PinIPv6AddressPool <Boolean> ] [-PinIPv6PAAAddressPool  
<Boolean> ] [-PinMACAddressPool <Boolean> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-  
VMSubnet <VMSubnet> ] [ <CommonParameters>]
```

### Detailed Description

The Set-SCVirtualNetworkAdapterConfiguration modifies the virtual network adapter configuration information that is contained within a virtual machine configuration.

### Parameters

#### **-IPv4Address<String>**

Specifies an IPv4 address.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-IPv4AddressPool<StaticIPAddressPool>**

Specifies a static address pool that contains IPv4 addresses.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-IPv4PAAddress<String>**

Specifies a provider IP address in IPv4 format.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-IPv4PAAddressPool<StaticIPAddressPool>**

Specifies a provider IP address pool that contains IP addresses in IPv4 format.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-IPv6Address<String>**

Specifies an IPv6 address.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-IPv6AddressPool<StaticIPAddressPool>**

Specifies a static address pool that contains IPv6 addresses.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-IPv6PAAddress<String>**

Specifies a provider IP address in IPv6 format.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-IPv6PAAddressPool<StaticIPAddressPool>**

Specifies a provider IP address pool that contains IP addresses in IPv6 format.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-MACAddress<String>**

Specifies the MAC address or a set of MAC addresses for a physical or virtual network adapter on a computer.

Example format for a single MAC address:

```
-MACAddress "00-15-5D-B4-DC-00"
```

Example formats for a set of MAC addresses:

```
-MACAddress "00-15-5D-B4-DC-00", "00-1A-A0-E3-75-29"
```

```
$Macs = "00-15-5D-B4-DC-00", "00-1A-A0-E3-75-29"
```

```
Set-SCPXEserver -MACAddress $Macs
```

NOTE: When used with New-SCPXEserver or Set-SCPXEserver, the MACAddress parameter updates the PXE interfaces from which the SCDM PXE Server listens for and responds to PXE requests.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-MACAddressPool<MACAddressPool>**

Specifies a MAC address pool.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PinIPv4AddressPool<Boolean>**

Indicates whether the IPv4 address pool chosen by the user is retained during service deployment configuration.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PinIPv4PAAddressPool<Boolean>**

Indicates whether the IPv4 provider address pool chosen by the user is retained during service deployment configuration.

---

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PinIPv6AddressPool<Boolean>**

Indicates whether the IPv6 address pool chosen by the user is retained during service deployment configuration.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PinIPv6PAAddressPool<Boolean>**

Indicates whether the IPv6 provider address pool chosen by the user is retained during service deployment configuration.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-PinMACAddressPool<Boolean>**

Indicates whether the MAC address pool chosen by the user is retained during service deployment configuration.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |



-

## **VirtualNetworkAdapterConfiguration<VirtualNetworkAdapterConfiguration>**

Specifies a virtual network adapter configuration object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-VMSubnet<VMSubnet>**

Specifies a VM subnet object.

To obtain a VM subnet object, use the Get-SCVMSubnet cmdlet.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **VirtualNetworkAdapterConfiguration**

---

## Examples

### 1: Set the properties of the virtual network adapter configuration for a virtual machine configuration.

The first command gets the service configuration object named Service01 and stores the object in the \$ServiceConfig variable.

The second command gets the computer tier configuration for the service configuration stored in \$ServiceConfig and stores the object in the \$TierConfig variable.

The third command gets the virtual machine configuration for the computer tier configuration stored in \$TierConfig and stores the object in the \$VMConfig variable.

The fourth command gets the virtual network adapter configuration for the first virtual machine configuration stored in \$VMConfig and stores the object in the \$VNAConfig variable.

The last command changes the IPv4Address property of the network adapter configuration stored in \$VNAConfig, and pins the value of the IP address, which prevents it from being changed during placement.

```
PS C:\> $ServiceConfig = Get-SCServiceConfiguration -Name "Service01"
PS C:\> $TierConfig = Get-SCComputerTierConfiguration -ServiceConfiguration $ServiceConfig
PS C:\> $VMConfig = Get-SCVMConfiguration -ComputerTierConfiguration $TierConfig
PS C:\> $VNAConfig = Get-SCVirtualNetworkAdapterConfiguration -VMConfiguration $VMConfig[0]
PS C:\> Set-SCVirtualNetworkAdapterConfiguration -VirtualNetworkAdapterConfiguration
$VNAConfig -IPv4Address "10.255.234.155" -PinIPv4AddressPool $True
```

## Related topics

[Get-SCComputerTierConfiguration](#)

[Get-SCServiceConfiguration](#)

[Get-SCVirtualNetworkAdapterConfiguration](#)

[Get-SCVMConfiguration](#)

---

# Set-SCVirtualNetworkAdapterNativePortProfile

---

## Set-SCVirtualNetworkAdapterNativePortProfile

Sets the properties of a virtual network adapter native port profile.

### Syntax

Parameter Set: Default

```
Set-SCVirtualNetworkAdapterNativePortProfile [-VirtualNetworkAdapterNativePortProfile]
<VirtualNetworkAdapterNativePortProfile> [-AllowIeeePriorityTagging <Boolean> ] [-
AllowMacAddressSpoofing <Boolean> ] [-AllowTeaming <Boolean> ] [-Description <String> ] [-
EnableDhcpGuard <Boolean> ] [-EnableIov <Boolean> ] [-EnableIPsecOffload <Boolean> ] [-
EnableRouterGuard <Boolean> ] [-EnableVmq <Boolean> ] [-JobVariable <String> ] [-
MaximumBandwidth <Int32> ] [-MinimumBandwidthAbsolute <Int32> ] [-MinimumBandwidthWeight
<Int32> ] [-Name <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer
<ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Set-SCVirtualNetworkAdapterNativePortProfile cmdlet updates the properties of a virtual network adapter native port profile.

### Parameters

#### -AllowIeeePriorityTagging<Boolean>

Specifies whether outgoing packets from the virtual machine adapter are allowed to be tagged with IEEE 802.1p priority. When set to \$false, the priority value in the packet is reset to zero.

The default value is \$False.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

### **-AllowMacAddressSpoofing<Boolean>**

Specifies whether MAC address spoofing is enabled. MAC address spoofing allows virtual machines to change the source MAC address in outgoing packets to one that is not assigned to them.

The default value is \$False.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-AllowTeaming<Boolean>**

Specifies whether guest teaming is allowed. Guest teaming allows the virtual network adapter to be teamed with other network adapters that are connected to the same switch.

The default value is \$False.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Description<String>**

States a description for the specified object.

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | false |
| Position? | named |

---

|                             |       |
|-----------------------------|-------|
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-EnableDhcpGuard<Boolean>**

Specifies whether DHCP guard is enabled. DHCP guard drops DHCP server messages from unauthorized virtual machines pretending to be DHCP servers.

The default value is \$False.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-EnableIov<Boolean>**

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-EnableIPsecOffload<Boolean>**

Specifies whether IPsec task offloading is enabled. When sufficient hardware resources are not available, the security associations are not offloaded and are handled in software by the guest operating system. Support from a physical network adapter and the guest operating system is required to offload IPsec tasks.

The default value is \$False.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-EnableRouterGuard<Boolean>**

Specifies whether router advertisement and redirection messages from authorized virtual machines pretending to be routers are dropped.

The default value is \$False.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-EnableVmq<Boolean>**

Specifies whether virtual machine queue (VMQ) is enabled. A physical network adapter is required to enable VMQ.

The default value is \$False.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-MaximumBandwidth<Int32>**

Specifies, in Megabits per second (Mbps), the maximum bandwidth for a virtual network adapter port profile. The default value is 0.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-MinimumBandwidthAbsolute<Int32>**

Specifies, in Megabits per second (Mbps), the minimum bandwidth for a virtual network adapter port profile. The default value is 0.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

### **-MinimumBandwidthWeight<Int32>**

Specifies the minimum bandwidth relative to how much bandwidth the virtual network adapter intends to use compared to other virtual network adapters connected to the same switch. Valid values are 0 to 100. The default value is 0.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Name<String>**

Specifies the name of a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                        |       |
|------------------------|-------|
| Aliases                | none  |
| Required?              | false |
| Position?              | named |
| Default Value          | none  |
| Accept Pipeline Input? | false |



|                             |       |
|-----------------------------|-------|
| Accept Wildcard Characters? | false |
|-----------------------------|-------|

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

-

## **VirtualNetworkAdapterNativePortProfile<VirtualNetworkAdapterNativePortProfile>**

Specifies a virtual network adapter native port profile object.

To obtain a virtual network adapter native port profile object, use the Get-SCVirtualNetworkAdapterNativePortProfile cmdlet.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

---

|                             |                |
|-----------------------------|----------------|
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Examples

### 1: Enable VMQ for a virtual network adapter native port profile.

This command gets the virtual network adapter native port profile object named VirtualNetworkAdapterNativePortProf01 and uses the pipeline operator to pass the object to Set-SCVirtualNetworkAdapterNativePortProfile which sets the EnableVMQ property to True for the virtual network adapter native port profile object.

```
PS C:\> Get-SCVirtualNetworkAdapterNativePortProfile -Name  
"VirtualNetworkAdapterNativePortProf01" | Set-SCVirtualNetworkAdapterNativePortProfile -  
EnableVMQ $True
```

### 2: Set the maximum and minimum bandwidth for a virtual network adapter native port profile.

This command gets the virtual network adapter native port profile object named VirtualNetworkAdapterNativePortProf01 and uses the pipeline operator to pass the object to Set-SCVirtualNetworkAdapterNativePortProfile which sets the maximum bandwidth for the object to 100 and the minimum bandwidth to 25.

```
PS C:\> Get-SCVirtualNetworkAdapterNativePortProfile -Name  
"VirtualNetworkAdapterNativePortProf01" | Set-SCVirtualNetworkAdapterNativePortProfile -  
MaximumBandwidth "100" -MinimumBandwidthAbsolute "25"
```

## Related topics

[Get-SCVirtualNetworkAdapterNativePortProfile](#)

[New-SCVirtualNetworkAdapterNativePortProfile](#)

[Remove-SCVirtualNetworkAdapterNativePortProfile](#)

---

# Set-SCVirtualNetworkAdapterPortProfileSet

---

## Set-SCVirtualNetworkAdapterPortProfileSet

Sets the properties of a virtual network adapter port profile set.

### Syntax

Parameter Set: Default

```
Set-SCVirtualNetworkAdapterPortProfileSet [-VirtualNetworkAdapterPortProfileSet]
<VirtualNetworkAdapterPortProfileSet> [[-Description] <String> ] [-
AddVirtualNetworkAdapterExtensionPortProfiles <VirtualNetworkAdapterExtensionPortProfile[]>
] [-IsDefaultPortProfileSet <Boolean> ] [-JobGroup <Guid> ] [-JobVariable <String> ] [-
LogicalSwitch <LogicalSwitch> ] [-Name <String> ] [-PortClassification <PortClassification>
] [-PROTipID <Guid> ] [-RemoveVirtualNetworkAdapterExtensionPortProfiles
<VirtualNetworkAdapterExtensionPortProfile[]> ] [-
RemoveVirtualNetworkAdapterNativePortProfile] [-RunAsynchronously] [-
VirtualNetworkAdapterNativePortProfile <VirtualNetworkAdapterNativePortProfile> ] [-
VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Set-SCVirtualNetworkAdapterPortProfileSet cmdlet updates the properties of a virtual network adapter port profile set object.

### Parameters

-

### AddVirtualNetworkAdapterExtensionPortProfiles<VirtualNetworkAdapterExtensionPortProfile[]>

Adds the specified array of virtual network adapter extension port profiles.

|               |       |
|---------------|-------|
| Aliases       | none  |
| Required?     | false |
| Position?     | named |
| Default Value | none  |

---

|                             |                |
|-----------------------------|----------------|
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-Description<String>**

States a description for the specified object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | 2     |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-IsDefaultPortProfileSet<Boolean>**

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobGroup<Guid>**

|                        |       |
|------------------------|-------|
| Aliases                | none  |
| Required?              | false |
| Position?              | named |
| Default Value          | none  |
| Accept Pipeline Input? | false |

---

|                             |       |
|-----------------------------|-------|
| Accept Wildcard Characters? | false |
|-----------------------------|-------|

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-LogicalSwitch<LogicalSwitch>**

Specifies a logical switch object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-Name<String>**

Specifies the name of a VMM object.

|                        |       |
|------------------------|-------|
| Aliases                | none  |
| Required?              | false |
| Position?              | named |
| Default Value          | none  |
| Accept Pipeline Input? | false |

|                             |       |
|-----------------------------|-------|
| Accept Wildcard Characters? | false |
|-----------------------------|-------|

### **-PortClassification<PortClassification>**

Specifies the port classification of the switch to which the virtual network adapter connects.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

-

### **RemoveVirtualNetworkAdapterExtensionPortProfiles<VirtualNetworkAdapterExtensionPortProfile[]>**

Removes the specified array of virtual network adapter extension port profiles.

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | false |
| Position? | named |

|                             |                |
|-----------------------------|----------------|
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-RemoveVirtualNetworkAdapterNativePortProfile**

Removes the specified virtual network adapter native port profile.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

-

### **VirtualNetworkAdapterNativePortProfile<VirtualNetworkAdapterNativePortProfile>**

Specifies a virtual network adapter native port profile object.

To obtain a virtual network adapter native port profile object, use the Get-SCVirtualNetworkAdapterNativePortProfile cmdlet.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

-

### **VirtualNetworkAdapterPortProfileSet<VirtualNetworkAdapterPortProfileSet>**

Specifies a virtual network adapter port profile set object.

To obtain a virtual network adapter port profile set object, use the Get-SCVirtualNetworkAdapterPortProfileSet cmdlet.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |



---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Examples

### 1: Remove a virtual network adapter native port profile from a virtual network adapter port profile set.

This command gets the virtual network adapter port profile set object named VirtualNetworkAdapterPortProfSet01 and uses the pipeline operator to pass the object to Set-SCVirtualNetworkAdapterPortProfileSet which removes the virtual network adapter native port profile from the virtual network adapter port profile set object.

```
PS C:\> Get-SCVirtualNetworkAdapterPortProfileSet -Name "VirtualNetworkAdapterPortProfSet01"
| Set-SCVirtualNetworkAdapterPortProfileSet -RemoveVirtualNetworkAdapterNativePortProfile
```

## Related topics

[Get-SCVirtualNetworkAdapterPortProfileSet](#)

[New-SCVirtualNetworkAdapterPortProfileSet](#)

[Remove-SCVirtualNetworkAdapterPortProfileSet](#)

---

# Set-SCVirtualScsiAdapter

---

## Set-SCVirtualScsiAdapter

Changes properties of a virtual SCSI adapter used in VMM.

### Syntax

Parameter Set: Default

```
Set-SCVirtualScsiAdapter [-VirtualScsiAdapter] <VirtualSCSIAdapter> [-AdapterID <Byte> ] [-JobGroup <Guid> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-ScsiControllerType {DefaultTypeNoType | LsiLogic | BusLogic | ParaVirtualSCSI | LsiLogicSAS}] [-ShareVirtualScsiAdapter <Boolean> ] [ <CommonParameters>]
```

### Detailed Description

The Set-SCVirtualScsiAdapter cmdlet changes one or more properties of a virtual SCSI adapter used in a Virtual Machine Manager (VMM) environment. Settings that you can modify include specifying whether or not a virtual SCSI adapter is shared and setting the adapter ID.

Note: Using the ShareVirtualScsiAdapter parameter to share a virtual SCSI adapter on a virtual machine in order to enable guest clustering is supported only if the virtual machine is deployed on a VMware ESX host. The SharedVirtualScsiAdapter parameter is not used for a virtual machine on a Hyper-V host because a virtual machine on a Hyper-V host uses iSCSI for shared storage.

Note: Set-SCVirtualScsiAdapter is not used for Citrix XenServer hosts because the SCSI adapter on Citrix XenServer virtual machines is not configurable.

### Parameters

#### **-AdapterID<Byte>**

Specifies the logical unit number, or LUN ID. Hyper-V and XenServer do not expose this value, and it cannot be changed. For a VMware ESX host, the default is 7 and cannot be changed.

|               |       |
|---------------|-------|
| Aliases       | none  |
| Required?     | false |
| Position?     | named |
| Default Value | none  |

---

|                             |       |
|-----------------------------|-------|
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobGroup<Guid>**

Specifies an identifier for a series of commands that will run as a set just before the final command that includes the same job group identifier runs.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | false |
| Position? | named |

|                             |       |
|-----------------------------|-------|
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-ScsiControllerType<VMSCSIControllerType>**

Specifies a SCSI controller type. Valid values are: DefaultType, NoType, LsiLogic, BusLogic, ParaVirtualSCSI, LsiLogicSAS.

The acceptable values for this parameter are:

|                   |  |
|-------------------|--|
| DefaultTypeNoType |  |
| LsiLogic          |  |
| BusLogic          |  |
| ParaVirtualSCSI   |  |
| LsiLogicSAS       |  |

|                        |       |
|------------------------|-------|
| Aliases                | none  |
| Required?              | false |
| Position?              | named |
| Default Value          | none  |
| Accept Pipeline Input? | false |

|                             |       |
|-----------------------------|-------|
| Accept Wildcard Characters? | false |
|-----------------------------|-------|

### **-ShareVirtualScsiAdapter<Boolean>**

Specifies that a virtual SCSI adapter will be shared so that it can be used in guest clustering.

TYPE OF HOST      USES THIS PARAMETER

-----      -----

Hyper-V host      No (for guest clustering, use iSCSI storage)

XenServer host      No (Xen VMs always have exactly one SCSI adapter)

Note: When sharing a SCSI controller on a virtual machine on an ESX host, VMM defaults the SCSI sharing policy on VMware to "physical."

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VirtualScsiAdapter<VirtualSCSIAdapter>**

Specifies a virtual SCSI adapter object for a virtual machine.

TYPE OF HOST      NUMBER OF VIRTUAL SCSI ADAPTERS

-----      -----

Hyper-V      Up to 4 synthetic virtual SCSI adapters per VM,  
and up to 64 devices per adapter

Supports a virtual disk drive size up to 2040 GB.

Does not support emulated virtual SCSI adapters.

VMware ESX      Up to 4 virtual SCSI adapters per VM,  
and up to 15 devices per adapter.

Supports a virtual disk drive size up to 2048 GB.

Citrix XenServer      Always 1 virtual SCSI adapter per VM,  
and up to 8 devices per adapter.

Supports a virtual disk drive size up to 2048 GB.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

---

|                             |                |
|-----------------------------|----------------|
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **VirtualSCSIAdapter**

## Notes

- Requires a VMM virtual SCSI adapter object, which can be retrieved by using the Get-SCVirtualScsiAdapter cmdlet.

## Examples

### 1: Share a specific virtual SCSI adapter on a virtual machine to enable it for guest clustering.

The first command gets the virtual machine object named VM01 and stores the object in the \$VM variable.

The second command gets the SCSI adapter object on VM01 and stores the object in the \$Adapter variable. This example assumes that VM01 has one virtual SCSI adapter. However, a virtual machine can have up to four virtual SCSI adapters attached.

The last command enables the virtual SCSI adapter object stored in \$Adapter and specifies that it is shared so that it can be used in guest clustering.

NOTE: Using the Shared parameter to share a virtual SCSI adapter on a virtual machine is supported only if the virtual machine is deployed on an ESX host. The Shared parameter is not used for a virtual machine a Hyper-V host because a virtual machine on a Hyper-V host uses iSCSI for shared storage. The Shared parameter is also not used for a virtual machine on a XenServer host because XenServer-based virtual machines always have exactly one SCSI adapter.

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM01"
```

```
PS C:\> $Adapter = Get-SCVirtualScsiAdapter -VM $VM
```

```
PS C:\> Set-SCVirtualSCSIAdapter -VirtualScsiAdapter $Adapter -ShareVirtualScsiAdapter $True
```

---

## Related topics

[Get-SCVirtualScsiAdapter](#)

[New-SCVirtualScsiAdapter](#)

[Remove-SCVirtualScsiAdapter](#)

---

# Set-SCVirtualSwitchExtension

---

## Set-SCVirtualSwitchExtension

Modifies a virtual switch extension.

### Syntax

Parameter Set: BySecureValue

```
Set-SCVirtualSwitchExtension [-VirtualSwitchExtension] <VirtualSwitchExtension> -  
InstallParameterName <String> -InstallParameterSecureValue <SecureString> [-JobVariable  
<String> ] [-PROTipID <Guid> ] [-RemoveInstallParameter] [-RunAsynchronously] [-VMMServer  
<ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ByValue

```
Set-SCVirtualSwitchExtension [-VirtualSwitchExtension] <VirtualSwitchExtension> -  
InstallParameterName <String> [-InstallParameterValue <String> ] [-JobVariable <String> ] [-  
PROTipID <Guid> ] [-RemoveInstallParameter] [-RunAsynchronously] [-VMMServer  
<ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Set-SCVirtualSwitchExtension cmdlet modifies a virtual switch extension.

### Parameters

#### **-InstallParameterName<String>**

Specifies the name of an install parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |



---

## **-InstallParameterSecureValue<SecureString>**

Specifies a secure string that contains the value for an install parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-InstallParameterValue<String>**

Specifies a value for an install parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RemoveInstallParameter**

Indicates that the install parameter is removed.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-VirtualSwitchExtension<VirtualSwitchExtension>**

Specifies a virtual switch extension object.

To get a virtual switch extension object, use the Get-SCVirtualSwitchExtension cmdlet.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## **Related topics**

[Get-SCVirtualSwitchExtension](#)

---

# Set-SCVirtualSwitchExtensionManager

---

## Set-SCVirtualSwitchExtensionManager

Sets the properties of a virtual switch extension manager.

### Syntax

Parameter Set: Default

```
Set-SCVirtualSwitchExtensionManager [-VirtualSwitchExtensionManager]
<VirtualSwitchExtensionManager> [-AddVMHostGroup <HostGroup[]> ] [-ConfigurationProvider
<ConfigurationProvider> ] [-JobVariable <String> ] [-Manufacturer <String> ] [-Model
<String> ] [-PROTipID <Guid> ] [-RemoveVMHostGroup <HostGroup[]> ] [-RunAsAccount
<RunAsAccount> ] [-RunAsynchronously] [-VirtualSwitchExtensionManagerConnectionString
<String> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Set-SCVirtualSwitchExtensionManager cmdlet updates the properties of a virtual switch extension manager.

### Parameters

#### **-AddVMHostGroup<HostGroup[]>**

Adds one or more host groups to an existing host group array or private cloud.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-ConfigurationProvider<ConfigurationProvider>**

Specifies a configuration provider object. A configuration provider is a plug-in to VMM that translates VMM PowerShell commands to API calls that are specific to a type of load balancer. If no configuration provider is specified, VMM uses the Manufacturer and Model information to choose an available configuration provider. If no configuration provider is found, the load balancer will not be added.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-Manufacturer<String>**

Specifies the name of the company that manufactured a physical device.

Valid characters include: letters (a-z), numbers (0-9), underscore (\_), hyphen(-), dot(.), and single quote(').

|               |       |
|---------------|-------|
| Aliases       | none  |
| Required?     | false |
| Position?     | named |
| Default Value | none  |

---

|                             |       |
|-----------------------------|-------|
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Model<String>**

Specifies the model of a physical device.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RemoveVMHostGroup<HostGroup[]>**

Removes one or more host groups from a host group array or private cloud.

|               |       |
|---------------|-------|
| Aliases       | none  |
| Required?     | false |
| Position?     | named |
| Default Value | none  |

|                             |       |
|-----------------------------|-------|
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsAccount<RunAsAccount>**

Specifies a Run As account that contains credentials with permission to perform this action.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VirtualSwitchExtensionManager<VirtualSwitchExtensionManager>**

Specifies a virtual switch extension manager object.

To obtain a virtual switch extension manager object, use the Get-SCVirtualSwitchExtensionManager cmdlet.

|           |      |
|-----------|------|
| Aliases   | none |
| Required? | true |

|                             |                |
|-----------------------------|----------------|
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-VirtualSwitchExtensionManagerConnectionString<String>**

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Related topics**

[Add-SCVirtualSwitchExtensionManager](#)

[Get-SCVirtualSwitchExtensionManager](#)



---

[Read-SCVirtualSwitchExtensionManager](#)

[Remove-SCVirtualSwitchExtensionManager](#)

[Test-SCVirtualSwitchExtensionManager](#)

---

# Set-SCVMCheckpoint

---

## Set-SCVMCheckpoint

Modifies the properties of a virtual machine checkpoint object in VMM.

### Syntax

Parameter Set: Default

```
Set-SCVMCheckpoint -VMCheckpoint <VMCheckpoint> [-Description <String> ] [-JobVariable <String> ] [-Name <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

### Detailed Description

The Set-SCVMCheckpoint cmdlet modifies the properties of a virtual machine checkpoint object in Virtual Machine Manager (VMM).

For information about creating VMM checkpoints, type: "Get-Help New-SCVMCheckpoint -detailed".

### Parameters

#### **-Description<String>**

States a description for the specified object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

---

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Name<String>**

Specifies the name of a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VMCheckpoint<VMCheckpoint>**

Specifies a VMM virtual machine checkpoint object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **VMCheckpoint**

## **Notes**

- Requires a VMM checkpoint object, which can be retrieved by using the Get-SCVMCheckpoint cmdlet.

---

## Examples

### 1: Set the description for all checkpoints to a specified string.

This command gets all existing checkpoint objects from the VMM database and updates the description for these checkpoints.

```
PS C:\> Get-SCVMCheckpoint | Set-SCVMCheckpoint -Description "All checkpoints created prior to upgrade"
```

### 2: Modify the name and description for all checkpoints.

This command gets all existing checkpoint objects and modifies the name and description for each object.

```
PS C:\> Get-VMCheckpoint | Set-VMCheckpoint -Name "Checkpoint Before Upgrade" -Description "Checkpoint was created prior to upgrade"
```

### 3: Modify a specific checkpoint in an array of checkpoints.

The first command gets the virtual machine object named VM03 and stores the object in the \$VM variable.

The last command modifies the description for the first checkpoint object in the VMCheckpoints array for VM03.

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM03"
```

```
PS C:\> Set-SCVMCheckpoint -VMCheckpoint $VM.VMCheckpoints[0] -Description "First Checkpoint Before Upgrade"
```

## Related topics

[Get-SCVMCheckpoint](#)

[New-SCVMCheckpoint](#)

[Restore-SCVMCheckpoint](#)

---

# Set-SCVMConfiguration

---

## Set-SCVMConfiguration

Modifies the virtual machine configuration for a computer tier.

### Syntax

Parameter Set: Update

```
Set-SCVMConfiguration [-VMConfiguration] <BaseVMConfiguration> [-AvailabilitySetNames <List<String>> ] [-CapabilityProfile <CapabilityProfile> ] [-ComputerName <String> ] [-CostCenter <String> ] [-Description <String> ] [-JobVariable <String> ] [-Name <String> ] [-NoConnectedHost] [-PinVMHost <Boolean> ] [-PinVMLocation <Boolean> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-Tag <String> ] [-VMLocation <String> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: Cloud

```
Set-SCVMConfiguration [-VMConfiguration] <BaseVMConfiguration> -Cloud <Cloud> [-AvailabilitySetNames <List<String>> ] [-CapabilityProfile <CapabilityProfile> ] [-ComputerName <String> ] [-CostCenter <String> ] [-Description <String> ] [-JobVariable <String> ] [-Name <String> ] [-NoConnectedHost] [-PinVMHost <Boolean> ] [-PinVMLocation <Boolean> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-Tag <String> ] [-VMLocation <String> ] [ <CommonParameters>]
```

Parameter Set: VMHost

```
Set-SCVMConfiguration [-VMConfiguration] <BaseVMConfiguration> -VMHost <Host> [-AvailabilitySetNames <List<String>> ] [-CapabilityProfile <CapabilityProfile> ] [-ComputerName <String> ] [-CostCenter <String> ] [-Description <String> ] [-JobVariable <String> ] [-Name <String> ] [-NoConnectedHost] [-PinVMHost <Boolean> ] [-PinVMLocation <Boolean> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-Tag <String> ] [-VMLocation <String> ] [ <CommonParameters>]
```

Parameter Set: VMHostGroup

```
Set-SCVMConfiguration [-VMConfiguration] <BaseVMConfiguration> [-AvailabilitySetNames <List<String>> ] [-CapabilityProfile <CapabilityProfile> ] [-ComputerName <String> ] [-CostCenter <String> ] [-Description <String> ] [-JobVariable <String> ] [-Name <String> ] [-NoConnectedHost] [-PinVMHost <Boolean> ] [-PinVMLocation <Boolean> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-Tag <String> ] [-VMHostGroup <HostGroup> ] [-VMLocation <String> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

---

## Detailed Description

The Set-SCVMConfiguration cmdlet modifies the virtual machine configuration for a computer tier. The virtual machine configuration describes how the virtual machine will be configured when the service is deployed.

## Parameters

### **-AvailabilitySetNames<List<String>>**

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-CapabilityProfile<CapabilityProfile>**

Specifies a capability profile object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Cloud<Cloud>**

Specifies a private cloud object.

|           |      |
|-----------|------|
| Aliases   | none |
| Required? | true |

|                             |       |
|-----------------------------|-------|
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-ComputerName<String>**

Specifies the name of a computer that VMM can uniquely identify on your network. Valid formats are: FQDN, IPv4 or IPv6 address, or NetBIOS name.

NOTE: See the examples for a specific cmdlet to determine how that cmdlet specifies the computer name.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-CostCenter<String>**

Specifies the cost center for a virtual machine so that you can collect data about the allocation of virtual machines (or resources allocated to virtual machines) to make use of in your billing system.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Description<String>**

States a description for the specified object.



---

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Name<String>**

Specifies the name of a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-NoConnectedHost**

Removes the host from a virtual machine configuration.

---

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PinVMHost<Boolean>**

Indicates whether the virtual machine host chosen by the user is retained during service deployment configuration.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PinVMLocation<Boolean>**

Indicates whether the virtual machine location chosen by the user is retained during service deployment configuration.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-Tag<String>**

Specifies a word or phrase to associate with an object so that you can search for all objects with the specified set of tags. You can search for a subset of tags, or you can search for the full set of tags.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-VMConfiguration<BaseVMConfiguration>**

Specifies a virtual machine configuration object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-VMHost<Host>**

Specifies a virtual machine host object. VMM supports Hyper-V hosts, VMware ESX hosts, and Citrix XenServer hosts.

For more information about each type of host, type: "Get-Help Add-SCVMHost -detailed". See the examples for a specific cmdlet to determine how that cmdlet uses this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VMHostGroup<HostGroup>**

Specifies a virtual machine host group object or an array of host group objects.

|                        |       |
|------------------------|-------|
| Aliases                | none  |
| Required?              | false |
| Position?              | named |
| Default Value          | none  |
| Accept Pipeline Input? | false |

---

|                             |       |
|-----------------------------|-------|
| Accept Wildcard Characters? | false |
|-----------------------------|-------|

### **-VMLocation<String>**

Specifies the path to a virtual machine.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

### **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **VMConfiguration**

---

## Examples

### 1. Update the virtual machine configuration for a machine tier prior to deploying the service.

The first command gets the service configuration object named Service01 and stores the object in the \$ServiceConfig variable.

The second command gets the computer tier configuration for the service configuration stored in \$ServiceConfig and stores the object in the \$TierConfig variable.

The third command gets the virtual machine configuration for the computer tier configuration stored in \$TierConfig and stores the variable in the \$VMConfig variable.

The last command sets the description property of the first virtual machine configuration object stored in \$VMConfig, and displays the properties of the virtual machine configuration to the user.

```
PS C:\> $ServiceConfig = Get-SCServiceConfiguration -Name "Service01"
PS C:\> $TierConfig = Get-SCComputerTierConfiguration -ServiceConfiguration $ServiceConfig
PS C:\> $VMConfig = Get-SCVMConfiguration -ComputerTierConfiguration $TierConfig
PS C:\> Set-SCVMConfiguration -VMConfiguration $VMConfig[0] -Description "This is the updated virtual machine configuration"
```

### 2. Configure the virtual machine configuration object for a machine tier to pin the host for a virtual machine.

The first command gets the service configuration object named Service01 and stores the object in the \$ServiceConfig variable.

The second command gets the computer tier configuration object for the service configuration stored in \$ServiceConfig and stores the object in the \$TierConfig variable.

The third command gets the virtual machine configuration objects for the computer tier configuration stored in \$TierConfig and stores the objects in the \$VMConfig variable.

The last command sets the PinVMHost property to True for the first virtual machine configuration object stored in \$VMConfig. Therefore, when the service is deployed, the host for the virtual machine created with this configuration will not be changed.

```
PS C:\> $ServiceConfig = Get-SCServiceConfiguration -Name "Service01"
PS C:\> $TierConfig = Get-SCComputerTierConfiguration -ServiceConfiguration $ServiceConfig
PS C:\> $VMConfig = Get-SCVMConfiguration -ComputerTierConfiguration $TierConfig
PS C:\> Set-SCVMConfiguration -VMConfiguration $VMConfig[0] -PinVMHost $True
```

### 3. Configure the virtual machine configuration object for a service in a private cloud.

The first command gets the service configuration object named Service01 and stores the object in the \$ServiceConfig variable.

---

The second command gets the computer tier configuration for the service configuration stored in `$ServiceConfig` and stores the object in the `$TierConfig` variable.

The third command gets the virtual machine configuration objects for the computer tier configuration stored in `$TierConfig` and stores the objects in the `$VMConfig` variable.

The fourth command gets the private cloud object named Production and stores the object in the `$Cloud` variable.

The last command updates the description for the second virtual machine configuration object stored in `$VMConfig` for the private cloud stored in `$Cloud`.

```
PS C:\> $ServiceConfig = Get-SCServiceConfiguration -Name "Service01"
```

```
PS C:\> $TierConfig = Get-SCComputerTierConfiguration -ServiceConfiguration $ServiceConfig
```

```
PS C:\> $VMConfig = Get-SCVMConfiguration -ComputerTierConfiguration $TierConfig
```

```
PS C:\> $Cloud = Get-SCCloud -Name "Production"
```

```
PS C:\> Set-SCVMConfiguration -VMConfiguration $VMConfig[1] -Cloud $Cloud -Description "This is the new virtual machine configuration"
```

## Related topics

[Get-SCComputerTierConfiguration](#)

[Get-SCVMConfiguration](#)

---

# Set-SCVMHost

---

## Set-SCVMHost

Changes properties of a virtual machine host.

### Syntax

Parameter Set: XenServerHost

```
Set-SCVMHost [-VMHost] <Host> [-AvailableForPlacement <Boolean> ] [-BMCAddress <String> ] [-BMCCustomConfigurationProvider <ConfigurationProvider> ] [-BMCPort <UInt32> ] [-BMCProtocol {None | IPMI | SMASH | Custom} ] [-BMCRunAsAccount <RunAsAccount> ] [-Certificate <ClientCertificate> ] [-CPUPercentageReserve <UInt16> ] [-Credential <PSCredential> ] [-Custom1 <String> ] [-Custom10 <String> ] [-Custom2 <String> ] [-Custom3 <String> ] [-Custom4 <String> ] [-Custom5 <String> ] [-Custom6 <String> ] [-Custom7 <String> ] [-Custom8 <String> ] [-Custom9 <String> ] [-Description <String> ] [-DiskSpaceReserveMB <UInt64> ] [-EnableLiveMigration <Boolean> ] [-EnableSecureMode <Boolean> ] [-JobGroup <Guid> ] [-JobVariable <String> ] [-LiveMigrationMaximum <UInt32> ] [-LiveStorageMigrationMaximum <UInt32> ] [-MaintenanceHost <Boolean> ] [-ManagementAdapterMACAddress <String> ] [-MaxDiskIOReservation <UInt64> ] [-MemoryReserveMB <UInt64> ] [-MigrationAuthProtocol {CredSSP | Kerberos} ] [-MigrationSubnet <String[]> ] [-NetworkPercentageReserve <UInt16> ] [-NumaSpanningEnabled <Boolean> ] [-OverrideHostGroupReserves <Boolean> ] [-PROTipID <Guid> ] [-RemoteConnectCertificatePath <String> ] [-RemoteConnectEnabled <Boolean> ] [-RemoteConnectPort <UInt32> ] [-RemoveRemoteConnectCertificate] [-RunAsynchronously] [-SecureRemoteConnectEnabled <Boolean> ] [-SMBiosGuid <Guid> ] [-TCPPort <UInt32> ] [-UseAnyMigrationSubnet <Boolean> ] [-VMHostManagementCredential <VMMCredential> ] [-VMPaths <String> ] [ <CommonParameters>]
```

### Detailed Description

The Set-SCVMHost cmdlet changes one or more properties of a virtual machine host managed by Virtual Machine Manager (VMM). Settings that you can modify with the Set-SCVMHost cmdlet are summarized as follows:

#### AVAILABILITY AS A HOST FOR VIRTUAL MACHINES

-----

You can specify whether a host is currently considered by the VMM placement process as a candidate on which to place virtual machines.

#### HOST RESERVE SETTINGS

-----

You can configure the following host reserve settings:

- Percentage of CPU usage to set aside for use by the host.



- Amount of disk space (MB) to set aside for use by the host.
- Maximum number of disk I/O operations per second (IOPS) to set aside for use by the host.
- Amount of memory (MB) to set aside for use by the host.
- Percentage of network capacity to set aside for use by the host.

The VMM placement process will not recommend placing a virtual machine on a host unless the resource requirements of the virtual machine can be met without using the host reserves. If you do not specify reserve settings, VMM uses default settings.

#### VIRTUAL MACHINE PATHS

-----

You can specify, as a set of default paths, locations on a host where virtual machine files can be stored.

#### CREDENTIAL FOR MANAGING HOSTS IN A PERIMETER NETWORK OR NON-TRUSTED DOMAIN

-----

You can specify the password for an account used to manage Hyper-V hosts that are located in a perimeter network or in a non-trusted domain.

#### REMOTE CONNECTION SETTINGS

-----

You can configure remote connection settings for Hyper-V hosts (VMConnect) that enable users to connect to virtual machines remotely. This setting does not apply to virtual machines on VMware ESX hosts.

## Parameters

### **-AvailableForPlacement<Boolean>**

Indicates whether the VMM placement process will consider this host or this volume on a host to be eligible as a possible location on which to deploy virtual machines. If this parameter is set to False, you can choose to deploy virtual machines on this host or volume anyway. The default value is True. This parameter does not apply to VMware ESX hosts.

When this parameter is used with network adapters, if set to \$False, then placement will not consider the logical networks configured on this network adapter to determine if the host is suitable for connecting a virtual machine.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-BMCAddress<String>**

Specifies, or updates, the out-of-band baseboard management controller (BMC) address for a specific physical machine. This might be an IP address, the fully qualified domain name (FQDN), or the DNS prefix (which is usually the same name as the NetBIOS name).

Typically, the BMC address and its connection to the network are separate from the IP address associated with a standard network adapter. Alternatively, some computers do use a standard network adapter to provide a single address for the BMC and for the network adapter. However, the BMC address has a unique port and is thus uniquely identifiable on the network.

Example IPv4 format: -BMCAddress "10.0.0.21"

Example Ipv6 format: -BMCAddress "2001:4898:2a:3:657b:9c7a:e1f0:6829"

Example FQDN format: -BMCAddress "Computer01.Contoso.com"

Example NetBIOS format: -BMCAddress "Computer01"

NOTE: By default, VMM uses an IP address or FQDN for the BMCAddress. However, it is also possible to create a Windows PowerShell module that enables you to specify other types of addresses as the BMC address.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-BMCCustomConfigurationProvider<ConfigurationProvider>**

Specifies, or updates, a configuration provider object for a baseboard management controller (BMC). A configuration provider is a plug-in to VMM that translates VMM PowerShell commands to API calls that are specific to a type of baseboard management controller. This parameter should be used with the Custom BMCProtocol.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-BMCPort<UInt32>**

Specifies, or updates, the out-of-band baseboard management controller (BMC) port for a specific physical machine. A BMC port is also known as a service processor port. Example default ports are 623 for IPMI and 443 for SMASH over WS-Man.

Example format: -BMCPort 80

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-BMCProtocol<OutOfBandManagementType>**

Specifies, or updates, the protocol that VMM uses to communicate with the out-of-band baseboard management controller (BMC). Valid values are: IPMI, SMASH, Custom.

A BMC (also known as a service processor or management controller) is a specialized controller on the motherboard of a server that acts an interface between the hardware and system management software. If the motherboard of a physical machine includes a BMC, when the machine is plugged in (whether it is powered off or powered on, and whether or not an operating system is installed), information about system hardware and the state of that system hardware health is available.

Example format: -BMCProtocol "Custom"

NOTE: The Custom protocol requires using the BMCCustomConfigurationProvider.

The acceptable values for this parameter are:

|        |  |
|--------|--|
| None   |  |
| IPMI   |  |
| SMASH  |  |
| Custom |  |

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | false |
| Position? | named |

---

|                             |       |
|-----------------------------|-------|
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-BMCRunAsAccount<RunAsAccount>**

Specifies the Run As account to use with the baseboard management controller (BMC) device.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Certificate<ClientCertificate>**

Specifies a security certificate object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-CPUPercentageReserve<UInt16>**

Specifies the percentage of CPU to reserve for the use of the operating system on the physical host computer. If you do not use this parameter to specify the reserve, the default setting for the host group is used: 10 percent. The VMM placement process will not recommend that a virtual machine be placed on a host unless its resource requirements can be met without using host reserves.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Credential<PSCredential>**

Specifies a credential object or, for some cmdlets, a Run As account object that contains the user name and password of an account that has permission to perform this action. Or, in the case of Restart-SCJob, has permission to complete a restarted task.

For more information about the PSCredential object, type: "Get-Help Get-Credential".

For more information about Run As accounts, type: "Get-Help New-SCRunAsAccount".

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Custom1<String>**

Specifies a custom property on a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-Custom10<String>**

Specifies a custom property on a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-Custom2<String>**

Specifies a custom property on a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-Custom3<String>**

Specifies a custom property on a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-Custom4<String>**

Specifies a custom property on a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-Custom5<String>**

Specifies a custom property on a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-Custom6<String>**

Specifies a custom property on a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-Custom7<String>**

Specifies a custom property on a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-Custom8<String>**

Specifies a custom property on a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-Custom9<String>**

Specifies a custom property on a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |



---

## **-Description<String>**

States a description for the specified object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-DiskSpaceReserveMB<UInt64>**

Specifies, in megabytes (MB), the amount of disk space to reserve for the use of the operating system on the physical host computer. If you do not use this parameter to specify the reserve, the default setting for the host group is used: 100 MB. The VMM placement process will not recommend that a virtual machine be placed on a host unless its resource requirements can be met without using host reserves.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-EnableLiveMigration<Boolean>**

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

### **-EnableSecureMode<Boolean>**

Indicates whether VMM communicates with VMware ESX hosts and Citrix XenServer hosts in secure mode. The default value is \$True.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobGroup<Guid>**

Specifies an identifier for a series of commands that will run as a set just before the final command that includes the same job group identifier runs.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                        |       |
|------------------------|-------|
| Aliases                | none  |
| Required?              | false |
| Position?              | named |
| Default Value          | none  |
| Accept Pipeline Input? | false |

---

|                             |       |
|-----------------------------|-------|
| Accept Wildcard Characters? | false |
|-----------------------------|-------|

### **-LiveMigrationMaximum<UInt32>**

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-LiveStorageMigrationMaximum<UInt32>**

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-MaintenanceHost<Boolean>**

This parameter is obsolete. Use AvailableForPlacement instead.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-ManagementAdapterMACAddress<String>**

Specifies the MAC address of the physical network adapter on the computer that is to be used by the VMM server to communicate with this host.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-MaxDiskIOReservation<UInt64>**

Specifies the maximum disk I/O per second (IOPS) on the physical host computer. If you do not use this parameter to specify the reserve, the default setting for the host group is used: 10000. The VMM placement process will not recommend that a virtual machine be placed on a host unless its resource requirements can be met without using host reserves.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-MemoryReserveMB<UInt64>**

Specifies, in megabytes (MB), the amount of memory to reserve for the use of the host operating system on the physical host computer. If you do not use this parameter to specify the reserve, the default setting for the host group is used: 256 MB. The VMM placement process will not recommend that a virtual machine be placed on a host unless its resource requirements can be met without using host reserves.

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | false |

|                             |       |
|-----------------------------|-------|
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-MigrationAuthProtocol<MigrationAuthProtocolType>**

The acceptable values for this parameter are:

|          |  |
|----------|--|
| CredSSP  |  |
| Kerberos |  |

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-MigrationSubnet<String[]>**

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-NetworkPercentageReserve<UInt16>**

Specifies the percentage of network capacity to reserve for the use of the host operating system on the physical host computer. If you do not use this parameter to specify the reserve, the default setting for the host group is used: 10 percent. The VMM placement process will not recommend that a virtual machine be placed on a host unless its resource requirements can be met without using host reserves.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-NumaSpanningEnabled<Boolean>**

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-OverrideHostGroupReserves<Boolean>**

Indicates, when set to \$True, that the host reserve settings from the parent host group will be overridden by the provided settings.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RemoteConnectCertificatePath<String>**

This parameter is obsolete.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RemoteConnectEnabled<Boolean>**

Enables, when set to \$True, a connection on a host server that lets users connect to their virtual machines remotely. This parameter only applies to virtual machines on Hyper-V hosts. It is not applicable to virtual machines on VMware ESX hosts or Citrix XenServer hosts.

|                        |       |
|------------------------|-------|
| Aliases                | none  |
| Required?              | false |
| Position?              | named |
| Default Value          | none  |
| Accept Pipeline Input? | false |

---

|                             |       |
|-----------------------------|-------|
| Accept Wildcard Characters? | false |
|-----------------------------|-------|

### **-RemoteConnectPort<UInt32>**

Specifies a default value for the TCP port to use when a remote user connects to a virtual machine. Typically, the default port for a Hyper-V host is 2179. This parameter does not apply to VMware ESX hosts or Citrix XenServer hosts.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RemoveRemoteConnectCertificate**

This parameter is obsolete.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | false |
| Position? | named |



|                             |       |
|-----------------------------|-------|
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-SecureRemoteConnectEnabled<Boolean>**

This parameter is obsolete.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-SMBiosGuid<Guid>**

Specifies the System Management BIOS globally unique identifier (SMBIOS GUID) for a physical computer that is associated with a record for that physical computer in VMM. SMBIOS defines data structures and access methods that enable a user or application to store and retrieve information about hardware on this computer, such as the name of the system, manufacturer, or the system BIOS version. Windows operating systems retrieve SMBIOS data at system startup and make that data available to programs.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-TCPPort<UInt32>**

Specifies a numeric value that represents a TCP port.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-UseAnyMigrationSubnet<Boolean>**

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMHost<Host>**

Specifies a virtual machine host object. VMM supports Hyper-V hosts, VMware ESX hosts, and Citrix XenServer hosts.

For more information about each type of host, type: "Get-Help Add-SCVMHost -detailed". See the examples for a specific cmdlet to determine how that cmdlet uses this parameter.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-VMHostManagementCredential<VMMCredential>**

This parameter is obsolete.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VMPaths<String>**

Specifies a set of default paths (as strings separated by the pipeline operator) on a host where virtual machine files can be stored.

Example format: -VMPaths "C:\Folder1|C:\Folder2|C:\Folder3"

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **Host**

## **Examples**

### **1: Make a host available for placement.**

The first command gets the host object named VMHost01 and stores the object in the \$VMHost variable.

---

The second command makes VMHost01 available as a host for virtual machines. Setting the parameter AvailableForPlacement to True enables the VMM placement process to evaluate this host as a possible candidate on which to deploy virtual machines.

```
PS C:\> $VMHost = Get-SCVMHost -ComputerName "VMHost01"  
PS C:\> Set-SCVMHost -VMHost $VMHost -AvailableForPlacement $True
```

## 2: Enable remote connections on a Hyper-V host.

The first command gets the host object named VMHost01 and stores the object in the \$VMHost variable.

The second command enables remote connections on VMHost01 and sets the port used for remote connections to 5900.

Enabling remote connections on a Hyper-V host allows users to remotely access and manage their virtual machines on the host.

```
PS C:\> $VMHost = Get-SCVMHost -ComputerName "VMHost01"  
PS C:\> Set-SCVMHost -VMHost $VMHost -RemoteConnectEnabled $True -RemoteConnectPort 5900
```

## 3. Update the virtual machine paths for a host.

The first command gets the host object named VMHost01 and stores the object in the \$VMHost variable.

The second command updates the VMPaths property for the host stored in \$VMHost by adding the path "D:\VirtualMachinePath" to the list of virtual machine paths on that host.

```
PS C:\> $VMHost = Get-SCVMHost -ComputerName "VMHost01"  
PS C:\> Set-SCVMHost -VMHost $VMHost -VMPaths "C:\ProgramData\Microsoft\Windows\Hyper-V\D:\VirtualMachinePath"
```

## 4: Update the resource reserves for a host.

The first command gets the host object named VMHost01 and stores the object in the \$VMHost variable.

The second command updates the specified properties for VMHost01.

```
PS C:\> $VMHost = Get-SCVMHost -ComputerName "VMHost01"  
PS C:\> Set-SCVMHost -VMHost $VMHost -CPUPercentageReserve 40 -DiskSpaceReserveMB 2048 -  
MaxDiskIOReservation 500 -MemoryReserveMB 1024 -NetworkPercentageReserve 40
```

---

## 5: Update the bare-metal computer username and password for a specified physical host.

The first command gets the host object named VMHost02 and stores the object in the \$VMHost variable.

The second command gets the Run As account object named BMCCreds and stores the object in the \$BMCRAA variable.

The third command updates the host stored in \$VMHost with the new Run As account stored in \$BMCRAA.

The last command refreshes the host stored in \$VMHost using its out-of-band interface, which updates the Run As account for the host.

```
PS C:\> $VMHost = Get-SCVMHost -ComputerName "VMHost02"
PS C:\> $BMCRAA = Get-SCRunAsAccount -Name "BMCCreds"
PS C:\> Set-SCVMHost $VMHost -BMCRUNAsAccount $BMCRAA
PS C:\> Read-SCVMHost -VMHost $VMHost -RefreshOutOfBandProperties
```

## 6: Update the certificates for XenServer hosts in a cluster.

The first command gets the host object named XenHost01 and stores the object in the \$VMHost variable.

The second gets the certificate object for XenHost01 and stores the object in the \$Certificate variable.

The last command uses the certificate supplied in \$Cert to enable VMM to communicate with XenHost01 in secure mode.

```
PS C:\> $VMHost = Get-VMHost -ComputerName "XenHost01"
PS C:\> $Cert = Get-SCCertificate -Computername $VMHost.Name
PS C:\> Set-SCVMHost -VMHost $VMHost -Certificate $Cert -EnableSecureMode $True
```

## Related topics

[Add-SCVMHost](#)

[Remove-SCVMHost](#)

---

# Set-SCVMHostCluster

---

## Set-SCVMHostCluster

Modifies the properties of a virtual machine host cluster managed by VMM.

### Syntax

Parameter Set: Default

```
Set-SCVMHostCluster [-VMHostCluster] <HostCluster> [-ClusterReserve <UInt32> ] [-Description <String> ] [-JobGroup <Guid> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-SetQuorumDisk <ClientObject> ] [-SetQuorumNodeMajority] [-VMHostManagementCredential <VMMCredential> ] [ <CommonParameters>]
```

### Detailed Description

Modifies the properties of a host cluster managed by Virtual Machine Manager (VMM). Properties that you can modify include changing the cluster reserve setting.

#### CLUSTER RESERVE

-----

The cluster reserve setting specifies the number of host failures that a host cluster can sustain before VMM considers the cluster to be over-committed. An over-committed host cluster is one that cannot withstand the specified number of host failures and still keep all of the virtual machines in the cluster running.

Virtual Machine Manager uses the following processes to determine over-commitment:

- Host Placement. The placement process calculates whether adding a new virtual machine to the host cluster will over-commit the host cluster and, if so, placement stops recommending the deployment of additional virtual machines on hosts in that cluster.
- Cluster Refresher. The host cluster refresher calculates, at periodic intervals, whether a host cluster is over-committed or not based on the following events:
  - A change in the value specified for the ClusterReserve parameter.
  - The failure or removal of nodes from the host cluster.
  - The addition of nodes to the host cluster.
  - The discovery of new virtual machines on nodes in the host cluster.

The following examples illustrate how over-commitment works:

- Example of over-commitment when all nodes are functioning:

---

If you specify a cluster reserve of 2 for an 8-node host cluster, and all 8 nodes are functioning, the host cluster is over-committed if any combination of 6 (8 minus 2) nodes lacks the capacity to accommodate existing virtual machines.

- Example of over-commitment when some nodes are not functioning:

If you specify a cluster reserve of 2 for an 8-node host cluster, but only 5 nodes are functioning, the host cluster is over-committed if any combination of 3 (5 minus 2) nodes lacks the capacity to accommodate existing virtual machines.

## Parameters

### **-ClusterReserve<UInt32>**

Specifies the number of host failures that a host cluster can sustain before VMM designates the cluster as over-committed. The default value is 1.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Description<String>**

States a description for the specified object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-JobGroup<Guid>**

Specifies an identifier for a series of commands that will run as a set just before the final command that includes the same job group identifier runs.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |



---

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-SetQuorumDisk<ClientObject>**

Specifies a disk to use as the quorum disk for the cluster.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-SetQuorumNodeMajority**

Sets the quorum mode to Node Majority for the cluster.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-VMHostCluster<HostCluster>**

Specifies a VMM host cluster object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-VMHostManagementCredential<VMMCredential>**

This parameter is obsolete.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **VMHostCluster**

---

## Examples

### 1: Change the setting for the cluster reserve for a host cluster.

The first command gets the host cluster object named VMHostCluster01.Contoso.com and stores the object in the \$VMHostCluster variable.

The second command changes value for the cluster reserve for host cluster VMHostCluster01 to 2.

```
PS C:\> $VMHostCluster = Get-SCVMHostCluster -Name "VMHostCluster01.Contoso.com"
```

```
PS C:\> Set-SCVMHostCluster -VMHostCluster $VMHostCluster -ClusterReserve 2
```

## Related topics

[Add-SCVMHostCluster](#)

[Get-SCVMHostCluster](#)

[Install-SCVMHostCluster](#)

[Move-SCVMHostCluster](#)

[Read-SCVMHostCluster](#)

[Remove-SCVMHostCluster](#)

[Test-SCVMHostCluster](#)

[Uninstall-SCVMHostCluster](#)

---

# Set-SCVMHostGroup

---

## Set-SCVMHostGroup

Changes the properties of a host group in VMM.

### Syntax

Parameter Set: Default

```
Set-SCVMHostGroup [-VMHostGroup] <HostGroup> [-Description <String> ] [-  
EnableUnencryptedFileTransfer <Boolean> ] [-InheritNetworkSettings <Boolean> ] [-JobGroup  
<Guid> ] [-JobVariable <String> ] [-Name <String> ] [-PROTipID <Guid> ] [-RunAsynchronously]  
[ <CommonParameters>]
```

### Detailed Description

The Set-SCVMHostGroup cmdlet changes one or more properties of a host group that contains hosts managed by Virtual Machine Manager (VMM). Properties that you can change include settings for name, description, and whether network settings are inherited from parent host groups.

### Parameters

#### **-Description<String>**

States a description for the specified object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-EnableUnencryptedFileTransfer<Boolean>**

Indicates, when set to True, that network file transfers do not require encryption. Allowing unencrypted network file transfers can improve performance if neither the source host nor the destination host requires encryption.

Use this parameter to:

- Enable unencrypted file transfers into, or out of, the library.
- Enable unencrypted file transfers into, out of, or within a host group.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-InheritNetworkSettings<Boolean>**

Specifies, when set to True, that the network settings for a host group will have the same values as those specified for its parent.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-JobGroup<Guid>**

Specifies an identifier for a series of commands that will run as a set just before the final command that includes the same job group identifier runs.

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | false |

---

|                             |       |
|-----------------------------|-------|
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Name<String>**

Specifies the name of a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | false |

|                             |       |
|-----------------------------|-------|
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VMHostGroup<HostGroup>**

Specifies a virtual machine host group object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- 
- **HostGroup**

## Notes

- Requires a VMM host group object, which can be retrieved by using the Get-SCVMHostGroup cmdlet.

## Examples

### 1: Change the inherit network settings property for an existing host group.

The first command gets the host group named HostGroup01 and stores it in the \$VMHostGroup variable.

The second command changes the value of the inherit network settings to false for the host group stored in \$VMHostGroup.

```
PS C:\> $VMHostGroup = Get-SCVMHostGroup -Name "VMHostGroup01"
```

```
PS C:\> Set-SCVMHostGroup -VMHostGroup $VMHostGroup -InheritNetworkSettings $False
```

## Related topics

[Get-SCVMHostGroup](#)

[Move-SCVMHostGroup](#)

[New-SCVMHostGroup](#)

[Remove-SCVMHostGroup](#)



---

# Set-SCVMHostNetworkAdapter

---

## Set-SCVMHostNetworkAdapter

Changes network-related properties for a physical network adapter on a host managed by VMM.

### Syntax

Parameter Set: LogicalNetwork

```
Set-SCVMHostNetworkAdapter [-VMHostNetworkAdapter] <HostNetworkAdapter> [-AddOrSetLogicalNetwork <LogicalNetwork> ] [-AvailableForPlacement <Boolean> ] [-Description <String> ] [-EnableNetworkVirtualization <Boolean> ] [-JobGroup <Guid> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RemoveLogicalNetwork <LogicalNetwork> ] [-RemoveUnassignedVLAN <UInt16[]> ] [-RemoveUplinkPortProfileSet] [-RunAsynchronously] [-SubnetVLAN <SubnetVLAN[]> ] [-UplinkPortProfileSet <UplinkPortProfileSet> ] [-UsedForManagement <Boolean> ] [-VLANMode {Access | Trunk | Invalid} ] [ <CommonParameters>]
```

Parameter Set: Manual

```
Set-SCVMHostNetworkAdapter [-VMHostNetworkAdapter] <HostNetworkAdapter> [-AvailableForPlacement <Boolean> ] [-Description <String> ] [-EnableNetworkVirtualization <Boolean> ] [-JobGroup <Guid> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RemoveUnassignedVLAN <UInt16[]> ] [-RemoveUplinkPortProfileSet] [-RunAsynchronously] [-UplinkPortProfileSet <UplinkPortProfileSet> ] [-UsedForManagement <Boolean> ] [-VLANEnabled <Boolean> ] [-VLANID <UInt16> ] [-VLANMode {Access | Trunk | Invalid} ] [-VLANTrunkID <UInt16[]> ] [ <CommonParameters>]
```

### Detailed Description

The Set-SCVMHostNetworkAdapter cmdlet changes network-related properties for a physical network adapter on a host managed by Virtual Machine Manager (VMM).

Properties that you can change with this cmdlet include:

\* VLAN SETTINGS: You can use the VLAN parameters to create or modify a single VLAN or multiple VLANs. For an illustration of how to specify VLAN settings, see the examples.

For more information about VLANs and additional examples that illustrate VLAN settings, type: "Get-Help Add-SCVMHostNetworkAdapter -detailed".

---

## Parameters

### **-AddOrSetLogicalNetwork<LogicalNetwork>**

Specifies a logical network that will be added or updated.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-AvailableForPlacement<Boolean>**

Indicates whether the VMM placement process will consider this host or this volume on a host to be eligible as a possible location on which to deploy virtual machines. If this parameter is set to False, you can choose to deploy virtual machines on this host or volume anyway. The default value is True. This parameter does not apply to VMware ESX hosts.

When this parameter is used with network adapters, if set to \$False, then placement will not consider the logical networks configured on this network adapter to determine if the host is suitable for connecting a virtual machine.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Description<String>**

States a description for the specified object.

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | false |

|                             |       |
|-----------------------------|-------|
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-EnableNetworkVirtualization<Boolean>**

Specifies whether network virtualization is enabled. The default value is \$False.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobGroup<Guid>**

Specifies an identifier for a series of commands that will run as a set just before the final command that includes the same job group identifier runs.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

---

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RemoveLogicalNetwork<LogicalNetwork>**

Specifies a logical network that will be removed.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RemoveUnassignedVlan<UInt16[]>**

Specifies that the specified VLANs will be removed from the VLAN trunk of the adapter if they are not associated with a logical network.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RemoveUplinkPortProfileSet**

Removes the uplink port profile set from the host network adapter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-SubnetVlan<SubnetVlan[]>**

Specifies one or more IP subnet and VLAN sets.

For information about creating a SubnetVlan, type: "Get-Help New-SCSubNetVlan".

---

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-UplinkPortProfileSet<UplinkPortProfileSet>**

Specifies an uplink port profile set object.

To obtain an uplink port profile set object, use the Get-SCUplinkPortProfileSet cmdlet.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-UsedForManagement<Boolean>**

Indicates whether the object is used to manage hosts.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-VlanEnabled<Boolean>**

Enables a virtual LAN (VLAN) for use by virtual machines on a Hyper-V or Citrix XenServer host.

Example format for a single VLAN: -VLANEnabled -VLANMode "Access" -VLANID 35

Example format for multiple VLANs: -VLANEnabled -VLANMode "Trunk" -VLANTrunkID  
1,2,100,200,1124

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VlanID<UInt16>**

Assigns a numerical identifier in the range 1-4094 to a virtual network adapter on a virtual machine or to a physical network adapter on a virtual machine host.

Configure a VlanID on a Hyper-V, VMware ESX, or Citrix XenServer host:

- On an externally bound physical network adapter when the Vlan mode is Access.

Configure a VlanID on a virtual network adapter of a virtual machine:

- Bound to a physical network adapter on the host, or
- Bound to an internal virtual network on the host.

Example format: -VlanEnabled

-VlanMode "Access" -VLANID 35

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-VlanMode<VlanMode>**

Specifies whether a virtual LAN (VLAN) on a virtual machine host supports traffic across a single VLAN ("Access" mode) or across multiple VLANs ("Trunk" mode). Valid values are: Access, Trunk.

The acceptable values for this parameter are:

|         |  |
|---------|--|
| Access  |  |
| Trunk   |  |
| Invalid |  |

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VlanTrunkID<UInt16[]>**

Assigns a list of numerical identifiers in the range 1-4094 to a physical network adapter on a Hyper-V host.

Example format: -VLANEnabled -VLANMode "Trunk" -VLANTrunkID 1,2,100,200,1124

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VMHostNetworkAdapter<HostNetworkAdapter>**

Specifies a physical network adapter object on a host to which virtual machines deployed on that host can connect.



---

Example format: -VMHostNetworkAdapter \$VMHostNIC

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **VMHostNetworkAdapter**

## Examples

### 1: Create a new virtual network on a host network adapter and specify a VLAN ID for the virtual network.

The first command gets the host object named VMHost02 and stores the object in the \$VMHost variable.

The second command gets the host network adapter object with a name that begins with "Intel(R) PRO/1000 on VMHost02 and stores the object in the \$HostAdapter variable.

The third command creates a virtual network named VirtualNetwork01 on VMHost02 that is bound to the host adapter stored in \$HostAdapter.

The last command enables a VLAN, sets the mode to Access (which routes traffic internally within a single VLAN), and assigns the network adapter a VLANID of 35.

NOTE: This example assumes that that your host is already connected to a VLAN or, if not, that your host has two network adapters. If your host has a single network adapter, assigning the adapter to a VLAN that is unavailable to the VMM server will prevent VMM from managing the host.

```
PS C:\> $VMHost = Get-SCVMHost -Computername "VMHost02.Contoso.com"
```

```
PS C:\> $HostAdapter = Get-SCVMHostNetworkAdapter -VMHost $VMHost | where {$_.Name -like "Intel(R) PRO/1000*" }
```

```
PS C:\> New-SCVirtualNetwork -Name "VirtualNetwork01" -VMHost $VMHost -VMHostNetworkAdapter $HostAdapter
```

---

```
PS C:\> Set-SCVMHostNetworkAdapter -VMHostNetworkAdapter $HostAdapter -VLanEnabled $True -
VLanMode "Access" -VLANID 35
```

## 2: Add VLAN tags to a host network adapter configured in "Trunk" mode.

The first command gets the host object named VMHost03 and stores the object in the \$VMHost variable.

The second command gets the host network adapter object by specifying the adapter name and stores the object in the \$VMHostNIC variable.

The third command uses the VlanTags property of the host network adapter object (\$VMHostNIC.VlanTags) and concatenates a new array of tags. The updated array retains the existing VlanTags and adds the listed tags to the array. The result of the concatenation is stored in \$NewVlanTags.

The last command passes the new list of VLAN tags to the VLANTrunkID parameter of Set-VMHostNetworkAdapter. The VLANMode parameter must specify the value "Trunk" whenever the VLANTrunkID parameter is used to modify the list of VLAN trunk numerical identifiers.

```
PS C:\> $VMHost = Get-SCVMHost -Computername "VMHost03.Contoso.com"
PS C:\> $VMHostNIC = Get-SCVMHostNetworkAdapter -VMHost $VMHost -Name "Adapter #3"
PS C:\> $NewVlanTags = $VMHostNIC.VlanTags + @(177,355,1012)
PS C:\> Set-SCVMHostNetworkAdapter -VMHostNetworkAdapter $VMHostNIC -VLanEnabled $TRUE -
VLanMode "Trunk" -VLANTrunkID $NewVlanTags
```

## Related topics

[Add-SCVMHostNetworkAdapter](#)

[Get-SCVMHostNetworkAdapter](#)

[New-SCVirtualNetwork](#)

[New-SCVirtualNetworkAdapter](#)

[Remove-SCVMHostNetworkAdapter](#)

[Set-SCVirtualNetwork](#)

[Set-SCVirtualNetworkAdapter](#)

---

# Set-SCVMHostProfile

---

## Set-SCVMHostProfile

Modifies the properties of a host profile.

### Syntax

Parameter Set: CreateNewHostProfileData

```
Set-SCVMHostProfile [[-Name] <String> ] -VMHostProfile <VMHostProfile> [-AnswerFile <Script> ] [-BypassVHDCConversion <Boolean> ] [-Description <String> ] [-DiskConfiguration <String> ] [-Domain <String> ] [-DomainJoinRunAsAccount <RunAsAccount> ] [-DriverMatchingTag <List<String>> ] [-FullName <String> ] [-GuiRunOnceCommands <String[]> ] [-JobVariable <String> ] [-LocalAdministratorCredential <VMCredentials> ] [-OrganizationName <String> ] [-Owner <String> ] [-ProductKey <String> ] [-PROTipID <Guid> ] [-RemoveAnswerFile] [-RemoveGuiRunOnceCommands] [-RunAsynchronously] [-TimeZone <Int32> ] [-VirtualHardDisk <VirtualHardDisk> ] [-VMHostNetworkAdapterProfile <VMHostNetworkAdapterProfile[]> ] [-VMMServer <ServerConnection> ] [-VMPATHS <String> ] [ <CommonParameters>]
```

### Detailed Description

The Set-SCVMHostProfile cmdlet modifies the properties of a host profile.

For information about how to create a host profile, type: "Get-Help New-SCVMHostProfile -detailed"

### Parameters

#### **-AnswerFile<Script>**

Specifies a script object stored in the VMM library to use as an answer file. The name of the answer file script depends on the operating system that you want to install on a virtual machine:

ANSWER FILE    GUEST OS TO INSTALL ON VM

-----

Sysprep.inf    Windows XP, Windows Server 2000, or Windows Server 2003

Unattend.xml    Windows Vista, Windows 7, or Windows Server 2008

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | false |
| Position? | named |

|                             |                |
|-----------------------------|----------------|
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-BypassVHDCConversion<Boolean>**

Indicates that a dynamic VHD attached to a host profile is not converted to a fixed type during deployment.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Description<String>**

States a description for the specified object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-DiskConfiguration<String>**

Specifies the disk and partition configuration for the physical computer that is to be deployed with Windows Hyper-V.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Domain<String>**

Specifies a fully qualified domain name (FQDN) for an Active Directory domain.

Example format: -Domain "Domain01.Corp.Contoso.com"

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-DomainJoinRunAsAccount<RunAsAccount>**

Specifies a Run As account that has permission to join the specified domain.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-DriverMatchingTag<List<String>>**

Specifies the custom tags to match with driver properties during deployment. When tags are specified, the deployment process matches the tags specified in the host profile with the tags in the drivers in order to download matching drivers and install them in the target image.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-FullName<String>**

Specifies the name of the person in whose name a virtual machine is registered.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-GuiRunOnceCommands<String[]>**

Specifies one or more commands to add to the [GuiRunOnce] section of an unattended answer file (such as Unattend.xml). Use single quotes around each string enclosed in double quotes.

Example format:

```
-GuiRunOnceCommands "C:\APF\APFPostSysPrepCopy.cmd PARAMS1",
"C:\APF\APFPostSysPrepCopy.cmd PARAMS1"
```

For information about how Windows PowerShell uses quotes, type: "Get-Help about\_Quoting\_Rules"

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-LocalAdministratorCredential<VMMSecret>**

Specifies the user name and password for the Local Administrator account (or Linux root account in the case of a Linux compatible Guest Operating System profile).

Specifying credentials on a new or existing template, on a new or existing guest operating system profile, or on a new virtual machine overrides any existing Administrator password.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Name<String>**

Specifies the name of a VMM object.

|               |       |
|---------------|-------|
| Aliases       | none  |
| Required?     | false |
| Position?     | 1     |
| Default Value | none  |

|                             |       |
|-----------------------------|-------|
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-OrganizationName<String>**

Specifies the name of the organization for the person in whose name a virtual machine is registered.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Owner<String>**

Specifies the owner of a VMM object in the form of a valid domain user account.

Example format: -Owner "Contoso\ReneeLo"

Example format: -Owner "ReneeLo@Contoso"

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-ProductKey<String>**

Specifies a product key. The product key is a 25-digit number that identifies the product license. A product key can be used to register VMM or an operating system to be installed on a virtual machine or host.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|



---

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RemoveAnswerFile**

Removes the answer file from host profile

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RemoveGuiRunOnceCommands**

Removes the GUI Run Once Commands property from host profile

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-TimeZone<Int32>**

Specifies a number (an index) that identifies a geographical region that shares the same standard time.

For a list of time zone indexes, see "Microsoft Time Zone Index Values" at:

<http://go.microsoft.com/fwlink/?LinkId=120935>. If no time zone is specified, the default time zone used for a virtual machine is the same time zone setting that is on the virtual machine host.

Example format to specify the GMT Standard Time zone: -TimeZone 085

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-VirtualHardDisk<VirtualHardDisk>**

Specifies a virtual hard disk object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VMHostNetworkAdapterProfile<VMHostNetworkAdapterProfile[]>**

Specifies a host network adapter profile object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VMHostProfile<VMHostProfile>**

Specifies a virtual machine host profile object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

---

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-VMPaths<String>**

Specifies a set of default paths (as strings separated by the pipeline operator) on a host where virtual machine files can be stored.

Example format: -VMPaths "C:\Folder1|C:\Folder2|C:\Folder3"

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **HostProfile[]**
- **HostProfile**

---

## Examples

### 1: Set the time zone value for a specified host profile.

This command gets the host profile object named HostProfile01 and sets its time zone value to the U.S. Eastern time zone (035).

```
PS C:\> Get-SCVMHostProfile -Name "HostProfile01" | Set-SCVMHostProfile -TimeZone 035
```

### 2: Add a host network adapter profile to a host profile.

The first command gets the host profile object named HostProfile01 and stores the object in the \$VMHostProfile variable.

The second command gets the host network adapter profile object with the ID of 259f47c7-c5a9-429d-a421-d232f9b34991 and stores the object in the \$HostNetAdapterProf variable.

The last command adds the host network adapter profile stored in \$HostNetAdapterProf to HostProfile01.

```
PS C:\> $VMHostProfile = Get-SCVMHostProfile -Name "HostProfile01"
```

```
PS C:\> $HostNetAdapterProf = Get-SCVMHostNetworkAdapterProfile -ID "259f47c7-c5a9-429d-a421-d232f9b34991"
```

```
PS C:\> Set-SCVMHostProfile -VMHostProfile $VMHostProfile -VMHostNetworkAdapterProfile $HostNetAdapterProf
```

## Related topics

[Get-SCVMHostProfile](#)

[New-SCVMHostProfile](#)

[Remove-SCVMHostProfile](#)

---

# Set-SCVMMServer

---

## Set-SCVMMServer

Changes properties of the VMM management server.

### Syntax

Parameter Set: Default

```
Set-SCVMMServer [-AutomaticLogicalNetworkCreationEnabled <Boolean> ] [-  
AutomaticVirtualNetworkCreationEnabled <Boolean> ] [-BackupLogicalNetworkMatch {Disabled |  
FirstDNSSuffixLabel | DNSSuffix | NetworkConnectionName | VirtualNetworkSwitchName} ] [-  
CEIPOptIn <Boolean> ] [-JobVariable <String> ] [-LibraryRefresherEnabled <Boolean> ] [-  
LibraryRefresherFrequency <UInt16> ] [-LogicalNetworkMatch {Disabled | FirstDNSSuffixLabel |  
DNSSuffix | NetworkConnectionName | VirtualNetworkSwitchName} ] [-PROTipID <Guid> ] [-  
RunAsynchronously] [-VMConnectDefaultPort <UInt32> ] [-VMMServer <ServerConnection> ] [-  
VMRCAccessAccount <String> ] [-VMRCDefaultPort <UInt32> ] [-VMSubnetIDRangeEnd <UInt32> ] [-  
VMSubnetIDRangeStart <UInt32> ] [ <CommonParameters>]
```

### Detailed Description

Changes one or more properties of the Virtual Machine Manager (VMM) management server. VMM settings that you can change with the Set-SCVMMServer cmdlet include the following:

Microsoft Customer Experience Improvement Program (CEIP) Participation

-----

You can use Set-SCVMMServer to specify whether to enable Service Quality Metrics (SQM) for this VMM management server.

Library Settings

-----

You can use Set-SCVMMServer to specify whether the Virtual Machine Manager library refresher is enabled and, if so, how often objects in the library are refreshed.

Remote Control

-----

You can use Set-SCVMMServer to configure the default remote control port to use when connecting to virtual machines (VMConnect).

Contact for Self-Service Users

-----

You can use Set-SCVMMServer to specify the e-mail address of an administrator who supports self-service users.

---

## Network Settings

-----  
You can use the Set-VMMServer cmdlet to configure how to match logical networks. You can also enable creation of logical and virtual networks automatically.

## Guest Agent Settings

-----  
You can use Set-SCVMMServer to select the method used to communicate with the VMM guest agent: FQDN or IP Address.

## Parameters

### **-AutomaticLogicalNetworkCreationEnabled<Boolean>**

Indicates whether logical networks are created automatically. When set to \$True, a new logical network is created, based on the logical network matching settings, if the host network network adapter is not associated with a logical network.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-AutomaticVirtualNetworkCreationEnabled<Boolean>**

Indicates whether virtual networks are created automatically. When set to \$True, if a host has one network adapter for placement associated with a logical network, but no virtual networks attached to the adapter, a new virtual network is created.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-BackupLogicalNetworkMatch<LogicalNetworkMatchOption>**

Specifies the network matching method to use if the primary network matching method is not available.  
The acceptable values for this parameter are:

|                          |  |
|--------------------------|--|
| Disabled                 |  |
| FirstDNSSuffixLabel      |  |
| DNSSuffix                |  |
| NetworkConnectionName    |  |
| VirtualNetworkSwitchName |  |

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-CEIPOptIn<Boolean>**

Enables Service Quality Metrics (SQM) for this server and thus participation in the Microsoft Customer Experience Improvement Program (CEIP), which collects information about problems customers encounter in order to improve the software in a later release.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.



---

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-LibraryRefresherEnabled<Boolean>**

Indicates, when set to \$True, that VMM library objects are refreshed.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-LibraryRefresherFrequency<UInt16>**

Specifies, in hours, the frequency at which objects in the VMM library are refreshed. The default setting is 1 hour.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-LogicalNetworkMatch<LogicalNetworkMatchOption>**

Specifies the logical network matching method to use for automatically creating logical networks. Valid values are: Disabled, FirstDNSSuffixLabel, DNSSuffix, NetworkConnectionName, VirtualNetworkSwitchName.

The acceptable values for this parameter are:

|                          |  |
|--------------------------|--|
| Disabled                 |  |
| FirstDNSSuffixLabel      |  |
| DNSSuffix                |  |
| NetworkConnectionName    |  |
| VirtualNetworkSwitchName |  |

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMConnectDefaultPort<UInt32>**

Specifies the default value for the TCP port used for Virtual Machine Connection (VMConnect) sessions on all Hyper-V hosts managed by this VMM server. Typically, the default port is 2179, but you can use this parameter to change the default. This parameter does not apply to VMware ESX Server hosts or Citrix XenServer hosts.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

---

## **-VMRCAccessAccount<String>**

Specifies the account to use when connecting to a virtual machine by using Virtual Machine Remote Control (VMRC).

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VMRCDefaultPort<UInt32>**

Specifies the default port to use when connecting to a virtual machine by using Virtual Machine Remote Control (VMRC).

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VMSubnetIDRangeEnd<UInt32>**

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-VMSubnetIDRangeStart<UInt32>**

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

### **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **VMMServer**

### **Examples**

#### **1: Set the frequency at which the library is refreshed.**

This command enables library refreshing for VMMServer01 and sets the refreshing frequency rate to every 24 hours.

```
PS C:\> Set-SCVMMServer -VMMServer "VMMServer01.Contoso.com" -LibraryRefresherEnabled $TRUE  
-LibraryRefresherFrequency 24
```

#### **2: Opt out of the Customer Experience Improvement Program.**

This command opts out of participation in the Microsoft Customer Experience Improvement Program (CEIP) by setting the CEIPOptIn parameter to \$False on VMMServer01.

```
PS C:\> Set-SCVMMServer -VMMServer "VMMServer01.Contoso.com" -CEIPOptIn $False
```

#### **3: Specify a self-service contact e-mail address.**

This command sets the self service contact e-mail address to "AdminHelp@Contoso.com" on VMMServer01.

---

```
PS C:\> Set-SCVMMServer -VMMServer "VMMServer01.Contoso.com" -SelfServiceContactEmail  
"AdminHelp@Contoso.com"
```

## Related topics

[Get-SCVMMServer](#)

---

# Set-SCVMNetwork

---

## Set-SCVMNetwork

Updates a VM network.

### Syntax

Parameter Set: Default

```
Set-SCVMNetwork [-VMNetwork] <VMNetwork> [-Description <String> ] [-JobVariable <String> ]  
[-Name <String> ] [-Owner <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-UserRole  
<UserRole> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Set-SCVMNetwork cmdlet updates the properties of a VM network.

To get a VM network object to update, use the Get-SCVMNetwork cmdlet.

For more information about VM networks, type Get-Help "New-SCVMNetwork -detailed".

### Parameters

#### **-Description<String>**

States a description for the specified object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

---

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Name<String>**

Specifies the name of a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Owner<String>**

Specifies the owner of a VMM object in the form of a valid domain user account.

Example format: -Owner "Contoso\ReneeLo"

Example format: -Owner "ReneeLo@Contoso"

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |



---

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-UserRole<UserRole>**

Specifies a user role object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-VMNetwork<VMNetwork>**

Specifies a VM network object.

To get a VM network object, use the Get-SCVMNetwork cmdlet.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## **Examples**

### **1: Add a description to a VM network.**

This command gets the VM network object named VMNetwork01 by using the Get-SCVMNetwork cmdlet and then uses the pipeline operator to pass the object to Set-SCVMNetwork which updates the description for the VM network object.

```
PS C:\> Get-SCVMNetwork -Name "VMNetwork01" | Set-SCVMNetwork -Description "The first VM network"
```

---

## Related topics

[Get-SCVMNetwork](#)

[New-SCVMNetwork](#)

[Remove-SCVMNetwork](#)

---

# Set-SCVMNetworkGateway

---

## Set-SCVMNetworkGateway

Sets the properties of a VM network gateway,

### Syntax

Parameter Set: Default

```
Set-SCVMNetworkGateway [-VMNetworkGateway] <VMNetworkGateway> [-Description <String> ] [-JobVariable <String> ] [-Name <String> ] [-NetworkRoutes <NetworkRoute[]> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Set-SCVMNetworkGateway cmdlet updates the properties of a VM network gateway,

### Parameters

#### -Description<String>

States a description for the specified object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### -JobVariable<String>

Specifies that job progress is tracked and stored in the variable named by this parameter.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Name<String>**

Specifies the name of a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-NetworkRoutes<NetworkRoute[]>**

Specifies an array of network routes.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-VMNetworkGateway<VMNetworkGateway>**

Specifies a VM network gateway object.

To obtain a VM network gateway object, use the Get-SCVMNetworkGateway cmdlet.

---

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

### Related topics

[Add-SCVMNetworkGateway](#)

[Get-SCVMNetworkGateway](#)

[Read-SCVMNetworkGateway](#)

[Remove-SCVMNetworkGateway](#)

---

# Set-SCVMSubnet

---

## Set-SCVMSubnet

Updates a VM subnet.

### Syntax

Parameter Set: Default

```
Set-SCVMSubnet [-VMSubnet] <VMSubnet> [-Description <String> ] [-JobVariable <String> ] [-MaxNumberOfPorts <UInt32> ] [-Name <String> ] [-PROTipID <Guid> ] [-RemoveMaxNumberOfPorts] [-RunAsynchronously] [-SubnetVLAN <SubnetVLAN[]> ] [-VMMServer <ServerConnection> ] [ <CommonParameters> ]
```

### Detailed Description

The Set-SCVMSubnet cmdlet updates the properties of a VM subnet.

To get a VM subnet object to update, use the Get-SCVMSubnet cmdlet.

For more information about VM subnets, type "Get-Help New-SCVMSubnet -detailed".

### Parameters

#### **-Description<String>**

States a description for the specified object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.



---

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-MaxNumberOfPorts<UInt32>**

Specifies the maximum number of ports supported by the VM subnet.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Name<String>**

Specifies the name of a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

---

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RemoveMaxNumberOfPorts**

Removes the value set on the VM subnet for the maximum number of ports supported by the VM subnet.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-SubnetVlan<SubnetVlan[]>**

Specifies one or more IP subnet and VLAN sets.

For information about creating a SubnetVlan, type: "Get-Help New-SCSubNetVlan".

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-VMSubnet<VMSubnet>**

Specifies a VM subnet object.

To obtain a VM subnet object, use the Get-SCVMSubnet cmdlet.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## Examples

### 1: Change the subnet VLAN for a VM subnet.

The first command creates a subnet VLAN object and stores the object in the \$SubnetVLAN variable.

The second command gets the VM subnet object named VMSubnet01 and uses the pipeline operator to pass the object to Set-SCVMSubnet which updates its subnet VLAN to 192.168.4.0/24.

```
PS C:\> $SubnetVLAN = New-SCSubnetVLAN -Subnet "192.168.4.0/24"
```

```
PS C:\> Get-SCVMSubnet -Name "VMSubnet01" | Set-SCVMSubnet -SubnetVLAN $SubnetVLAN
```

## Related topics

[Get-SCVMSubnet](#)

[New-SCVMSubnet](#)

[Remove-SCVMSubnet](#)

---

# Set-SCVMTemplate

---

## Set-SCVMTemplate

Changes properties of a virtual machine template used in VMM.

### Syntax

Parameter Set: Default

```
Set-SCVMTemplate [-VMTemplate] <Template> [-AnswerFile <Script> ] [-ApplicationProfile <ApplicationProfile> ] [-AutoLogonCount <UInt32> ] [-AutoLogonCredential <RunAsAccount> ] [-BlockDynamicOptimization <Boolean> ] [-BootOrder {Floppy | CD | IdeHardDrive | PxeBoot | Unknown} ] [-CapabilityProfile <CapabilityProfile> ] [-ComputerName <String> ] [-CostCenter <String> ] [-CPUCount <Byte> ] [-CPUExpectedUtilizationPercent <Int32> ] [-CPULimitForMigration <Boolean> ] [-CPULimitFunctionality <Boolean> ] [-CPUMaximumPercent <Int32> ] [-CPUPerVirtualNumaNodeMaximum <Byte> ] [-CPURelativeWeight <Int32> ] [-CPUReserve <Int32> ] [-CPUType <ProcessorType> ] [-Custom1 <String> ] [-Custom10 <String> ] [-Custom2 <String> ] [-Custom3 <String> ] [-Custom4 <String> ] [-Custom5 <String> ] [-Custom6 <String> ] [-Custom7 <String> ] [-Custom8 <String> ] [-Custom9 <String> ] [-Description <String> ] [-DisableAutoLogon] [-DiskIops <Int32> ] [-Domain <String> ] [-DomainJoinCredential <VMMCredential> ] [-DomainJoinOrganizationalUnit <String> ] [-DRProtectionRequired <Boolean> ] [-DynamicMemoryBufferPercentage <Int32> ] [-DynamicMemoryEnabled <Boolean> ] [-DynamicMemoryMaximumMB <Int32> ] [-DynamicMemoryMinimumMB <UInt16> ] [-Enabled <Boolean> ] [-FullName <String> ] [-GuiRunOnceCommands <String[]> ] [-HAVMPriority <UInt32> ] [-HighlyAvailable <Boolean> ] [-JobGroup <Guid> ] [-JobVariable <String> ] [-LinuxAdministratorSSHKey <SSHKey> ] [-LinuxDomainName <String> ] [-LocalAdministratorCredential <VMMCredential> ] [-MemoryMB <Int32> ] [-MemoryPerVirtualNumaNodeMaximumMB <Int32> ] [-MemoryWeight <Int32> ] [-MergeAnswerFile <Boolean> ] [-MonitorMaximumCount <Int32> ] [-MonitorMaximumResolution <String> ] [-NetworkUtilizationMbps <Int32> ] [-NumaIsolationRequired <Boolean> ] [-NumLock <Boolean> ] [-OperatingSystem <OperatingSystem> ] [-OrganizationName <String> ] [-Owner <String> ] [-ProductKey <String> ] [-PROTipID <Guid> ] [-QuotaPoint <UInt32> ] [-RemoveCapabilityProfile] [-RemoveDomainJoinOrganizationalUnit] [-RemoveServerFeatures] [-RunAsynchronously] [-SQLProfile <SQLProfile> ] [-Tag <String> ] [-TimeZone <Int32> ] [-UnattendSettings <Dictionary<String>> ] [-UserRole <UserRole> ] [-VirtualNumaNodesPerSocketMaximum <Byte> ] [-VirtualVideoAdapterEnabled <Boolean> ] [-Workgroup <String> ] [ <CommonParameters>]
```

Parameter Set: SingleVM

```
Set-SCVMTemplate [-VMTemplate] <Template> [-AnswerFile <Script> ] [-ApplicationProfile <ApplicationProfile> ] [-AutoLogonCount <UInt32> ] [-AutoLogonCredential <RunAsAccount> ] [-BlockDynamicOptimization <Boolean> ] [-BootOrder {Floppy | CD | IdeHardDrive | PxeBoot | Unknown} ] [-CapabilityProfile <CapabilityProfile> ] [-ComputerName <String> ] [-CostCenter <String> ] [-CPUCount <Byte> ] [-CPUExpectedUtilizationPercent <Int32> ] [-CPULimitForMigration <Boolean> ] [-CPULimitFunctionality <Boolean> ] [-CPUMaximumPercent <Int32> ] [-CPUPerVirtualNumaNodeMaximum <Byte> ] [-CPURelativeWeight <Int32> ] [-CPUReserve <Int32> ] [-CPUType <ProcessorType> ] [-Custom1 <String> ] [-Custom10 <String> ] [-Custom2 <String> ] [-Custom3 <String> ] [-Custom4 <String> ] [-Custom5 <String> ] [-Custom6 <String> ] [-Custom7 <String> ] [-Custom8 <String> ] [-Custom9 <String> ] [-Description <String> ] [-
```

---

```
DisableAutoLogon] [-DiskIops <Int32> ] [-Domain <String> ] [-DomainJoinCredential
<VMCredentials> ] [-DomainJoinOrganizationalUnit <String> ] [-DRProtectionRequired <Boolean>
] [-DynamicMemoryBufferPercentage <Int32> ] [-DynamicMemoryEnabled <Boolean> ] [-
DynamicMemoryMaximumMB <Int32> ] [-DynamicMemoryMinimumMB <UInt16> ] [-Enabled <Boolean> ]
[-FullName <String> ] [-GuiRunOnceCommands <String[]> ] [-HAVMPriority <UInt32> ] [-
HighlyAvailable <Boolean> ] [-JobGroup <Guid> ] [-JobVariable <String> ] [-
LinuxAdministratorSSHKey <SSHKey> ] [-LinuxDomainName <String> ] [-
LocalAdministratorCredential <VMCredentials> ] [-MemoryMB <Int32> ] [-
MemoryPerVirtualNumaNodeMaximumMB <Int32> ] [-MemoryWeight <Int32> ] [-MergeAnswerFile
<Boolean> ] [-MonitorMaximumCount <Int32> ] [-MonitorMaximumResolution <String> ] [-Name
<String> ] [-NetworkUtilizationMbps <Int32> ] [-NumaIsolationRequired <Boolean> ] [-NumLock
<Boolean> ] [-OperatingSystem <OperatingSystem> ] [-OrganizationName <String> ] [-Owner
<String> ] [-ProductKey <String> ] [-PROTipID <Guid> ] [-QuotaPoint <UInt32> ] [-
RemoveCapabilityProfile] [-RemoveDomainJoinOrganizationalUnit] [-RemoveServerFeatures] [-
RunAsynchronously] [-SQLProfile <SQLProfile> ] [-Tag <String> ] [-TimeZone <Int32> ] [-
UnattendSettings <Dictionary<String>> ] [-UserRole <UserRole> ] [-
VirtualNumaNodesPerSocketMaximum <Byte> ] [-VirtualVideoAdapterEnabled <Boolean> ] [-
Workgroup <String> ] [ <CommonParameters>]
```

## Detailed Description

Changes one or more properties of a virtual machine template used in a Virtual Machine Manager (VMM) environment.

Properties that you can change include settings for the name of the template, its description, owner, organization name, time zone, operating system, and product key; BIOS boot order; CPU number and characteristics; the amount memory on the host that is assigned to a virtual machine; the amount of bandwidth on the host's network available to a virtual machine; whether a virtual machine created with this template will be highly available (i.e., deployed on a host that is part of a host cluster); values for domain or workgroup as well as domain credentials; a point quota that specifies how many new virtual machines self-service users can create; custom fields; and other options.

Changes that you make to a template affect only the template. Changes do not affect any existing virtual machines that were previously created by using this template.

If you want to change the properties of a virtual disk drive, virtual floppy drive, virtual DVD drive, virtual network adapter, virtual SCSI adapter or virtual COM port associated with a specific template, you can use Set-SCVirtualDiskDrive, Set-SCVirtualFloppyDrive, Set-SCVirtualDVDDrive, Set-SCVirtualNetworkAdapter, Set-SCVirtualScsiAdapter, or Set-SCVirtualCOMPort cmdlets, respectively.

## Parameters

### **-AnswerFile<Script>**

Specifies a script object stored in the VMM library to use as an answer file. The name of the answer file script depends on the operating system that you want to install on a virtual machine:

```
ANSWER FILE GUEST OS TO INSTALL ON VM
```

---

-----

Sysprep.inf Windows XP, Windows Server 2000, or Windows Server 2003

Unattend.xml Windows Vista, Windows 7, or Windows Server 2008

|                             |                |
|-----------------------------|----------------|
| Aliases                     | SysPrepFile    |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-ApplicationProfile<ApplicationProfile>**

Specifies an application profile object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-AutoLogonCount<UInt32>**

Specifies the number of times that Windows should automatically log the administrator denoted in the Answer File on to the console session.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-AutoLogonCredential<RunAsAccount>**

Specifies the Run As account contained in the Answer File that Windows uses to log on to the console session when automatic administrator logon is enabled.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-BlockDynamicOptimization<Boolean>**

Indicates whether dynamic optimization is blocked for a virtual machine.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-BootOrder<BootDevice[]>**

Specifies the order of devices that a virtual machine on a Hyper-V host uses to start up. Valid values are: CD, IDEHardDrive, PXEBoot, Floppy.

Example format: -BootOrder PXEBoot,IDEHardDrive,CD,Floppy

The acceptable values for this parameter are:

|              |  |
|--------------|--|
| Floppy       |  |
| CD           |  |
| IdeHardDrive |  |
| PxeBoot      |  |
| Unknown      |  |



|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-CapabilityProfile<CapabilityProfile>**

Specifies a capability profile object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-ComputerName<String>**

Specifies the name of a computer that VMM can uniquely identify on your network. Valid formats are: FQDN, IPv4 or IPv6 address, or NetBIOS name.

NOTE: See the examples for a specific cmdlet to determine how that cmdlet specifies the computer name.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-CostCenter<String>**

Specifies the cost center for a virtual machine so that you can collect data about the allocation of virtual machines (or resources allocated to virtual machines) to make use of in your billing system.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-CPUCount<Byte>**

Specifies the number of CPUs on a virtual machine, on a hardware profile, or on a template. See the examples for a specific cmdlet to determine how that cmdlet uses this parameter.

TYPE OF HOST    NUMBER OF PROCESSORS

-----    -----

Hyper-V        Up to 4 CPUs per VM; varies by guest OS

VMware ESX    Up to 4 CPUs per VM for any supported guest OS

Exception: 1 CPU on a VM running Windows NT 4.0

Citrix XenServer Up to 8 CPUs per VM; varies by guest OS

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-CPUExpectedUtilizationPercent<Int32>**

Specifies the percent of CPU on the host that you expect this virtual machine to use. This value is used only when VMM determines a suitable host for the virtual machine.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-CPULimitForMigration<Boolean>**

Limits, when set to \$True, processor features for the specified virtual machine in order to enable migration to a physical computer that has a different version of the same processor as the source computer. VMM does not support migrating virtual machines between physical computers that have processors from different manufacturers.

|                             |                      |
|-----------------------------|----------------------|
| Aliases                     | LimitCPUForMigration |
| Required?                   | false                |
| Position?                   | named                |
| Default Value               | none                 |
| Accept Pipeline Input?      | false                |
| Accept Wildcard Characters? | false                |

### **-CPULimitFunctionality<Boolean>**

Enables running an older operating system (such as Windows NT 4.0) on a virtual machine deployed on a Hyper-V host or on a VMware ESX host by providing limited CPU functionality for the virtual machine.

|                             |                       |
|-----------------------------|-----------------------|
| Aliases                     | LimitCPUFunctionality |
| Required?                   | false                 |
| Position?                   | named                 |
| Default Value               | none                  |
| Accept Pipeline Input?      | false                 |
| Accept Wildcard Characters? | false                 |

---

## **-CPUMaximumPercent<Int32>**

Specifies the highest percentage of the total resources of a single CPU on the host that can be used by a specific virtual machine at any given time.

Example: -CPUMaximumPercent 80 (to specify 80 per cent)

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-CPUPerVirtualNumaNodeMaximum<Byte>**

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-CPURelativeWeight<Int32>**

Specifies the amount of CPU resources on a host that this virtual machine can use relative to other virtual machines on the same host. A virtual machine with a higher setting is allocated more CPU resources than a virtual machine with a lower setting.

TYPE OF HOST    RANGE OF RELATIVE VALUES

-----    -----

Hyper-V        1 to 10000

VMware ESX    2000 = High

1500 = Above Normal

1000 = Normal (default)

750 = Below Normal

500 = Low

---

1 to 1000000 = Custom

The VMware term for these values is "shares."

Citrix XenServer 1 to 65536, normal is 256

NOTE: See the examples for a specific cmdlet to determine how that cmdlet uses this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-CPUReserve<Int32>**

Specifies the minimum percentage of the resources of a single CPU on the host to allocate to a virtual machine. The percentage of CPU capacity that is available to the virtual machine is never less than this percentage.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-CPUType<ProcessorType>**

Specifies the type of CPU for a virtual machine. To retrieve a list of all CPU types that are available for use in virtual machines in a VMM environment, type: "Get-SCCPUType"

|               |       |
|---------------|-------|
| Aliases       | none  |
| Required?     | false |
| Position?     | named |
| Default Value | none  |

---

|                             |       |
|-----------------------------|-------|
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Custom1<String>**

Specifies a custom property on a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Custom10<String>**

Specifies a custom property on a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Custom2<String>**

Specifies a custom property on a VMM object.

|               |       |
|---------------|-------|
| Aliases       | none  |
| Required?     | false |
| Position?     | named |
| Default Value | none  |

---

|                             |       |
|-----------------------------|-------|
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Custom3<String>**

Specifies a custom property on a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Custom4<String>**

Specifies a custom property on a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Custom5<String>**

Specifies a custom property on a VMM object.

|               |       |
|---------------|-------|
| Aliases       | none  |
| Required?     | false |
| Position?     | named |
| Default Value | none  |

---

|                             |       |
|-----------------------------|-------|
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Custom6<String>**

Specifies a custom property on a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Custom7<String>**

Specifies a custom property on a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Custom8<String>**

Specifies a custom property on a VMM object.

|               |       |
|---------------|-------|
| Aliases       | none  |
| Required?     | false |
| Position?     | named |
| Default Value | none  |



---

|                             |       |
|-----------------------------|-------|
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Custom9<String>**

Specifies a custom property on a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Description<String>**

States a description for the specified object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-DisableAutoLogon**

Disables automatic administrator logon.

|               |       |
|---------------|-------|
| Aliases       | none  |
| Required?     | false |
| Position?     | named |
| Default Value | none  |

|                             |       |
|-----------------------------|-------|
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-DiskIops<Int32>**

Specifies the number of disk input/output operations per second (IOPS) on the host that can be used by a specific virtual machine.

Example: -DiskIO 1500 (to specify 1500 IOPS).

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Domain<String>**

Specifies a fully qualified domain name (FQDN) for an Active Directory domain.

Example format: -Domain "Domain01.Corp.Contoso.com"

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-DomainJoinCredential<VMMCredential>**

Specifies the user name and password of an account with permission to join a computer to the domain. A limited rights account should be used for joining computers (either physical or virtual) to the domain.

Example format for a PS credential:

\$DomainJoinCredential = Get-Credential

---

-Domain "ThisDomain.Corp.Contoso.com" -DomainJoinCredential \$DomainJoinCredential

Example format for a Run As account:

```
$DomainJoinCredential = Get-SCRunAsAccount -Name "RunAsAcct01"
```

-Domain "ThisDomain.Corp.Contoso.com" -DomainJoinCredential \$DomainJoinCredential

NOTE: You can use the DomainJoinCredential parameter to specify credentials (on a VMHostProfile) for joining a physical host computer to the domain, or to specify credentials (on a new or existing template, on a new or existing guest operating system profile, or on a new virtual machine) for joining a virtual machine to the domain.

|                             |                      |
|-----------------------------|----------------------|
| Aliases                     | JoinDomainCredential |
| Required?                   | false                |
| Position?                   | named                |
| Default Value               | none                 |
| Accept Pipeline Input?      | false                |
| Accept Wildcard Characters? | false                |

### **-DomainJoinOrganizationalUnit<String>**

Specifies the organizational unit to which the computer is joined during an unattended mini-setup.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-DRProtectionRequired<Boolean>**

This parameter is reserved for future use.

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | false |
| Position? | named |

|                             |       |
|-----------------------------|-------|
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-DynamicMemoryBufferPercentage<Int32>**

Specifies the percentage of memory above a virtual machine's current memory allocation which the host should try to reserve as a buffer. The default value is 20.

Example format: -DynamicMemoryTargetBufferPercentage 20

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-DynamicMemoryEnabled<Boolean>**

Enables, when set to \$True, dynamic memory for virtual machines. You can enable dynamic memory directly on a virtual machine, or on a template or hardware profile that will be used to create virtual machines. The default value is False.

REQUIRED: You can enable dynamic memory for a virtual machine only if that virtual machine is deployed on a host running Windows Server 2008 SP1 or later or if the virtual machine is stored in a library in a stopped state (hardware changes to a stored virtual machine can only be made if the virtual machine does not have snapshots). Enabling dynamic memory on a virtual machine stored in a library will limit placement of that machine to hosts running Windows Server 2008 SP1 or later.

Example format: -DynamicMemoryEnabled \$True

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-DynamicMemoryMaximumMB<Int32>**

Specifies the maximum amount of memory that can be allocated to a virtual machine if dynamic memory is enabled. The default value is 65536.

REQUIRED: You can enable dynamic memory for a virtual machine only if that virtual machine is deployed on a host running Windows Server 2008 R2 SP1 or later or if the virtual machine is stored in a library in a stopped state (hardware changes to a stored virtual machine can only be made if the virtual machine does not have snapshots). Enabling dynamic memory on a virtual machine stored in a library will limit placement of that machine to hosts running Windows Server 2008 R2 SP1 or later.

Example format: -DynamicMemoryMaximumMB 1024

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-DynamicMemoryMinimumMB<UInt16>**

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-Enabled<Boolean>**

Enables an object when set to \$True, or disables an object when set to \$False. For example, if you want to upgrade software on a virtual machine template, you can disable the template object in the VMM library to temporarily prevent users from using that object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-FullName<String>**

Specifies the name of the person in whose name a virtual machine is registered.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-GuiRunOnceCommands<String[]>**

Specifies one or more commands to add to the [GuiRunOnce] section of an unattended answer file (such as Unattend.xml). Use single quotes around each string enclosed in double quotes.

Example format:

```
-GuiRunOnceCommands "C:\APF\APFPostSysPrepCopy.cmd PARAMS1",
"C:\APF\APFPostSysPrepCopy.cmd PARAMS1"
```

For information about how Windows PowerShell uses quotes, type: "Get-Help about\_Quoting\_Rules"

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-HAVMPriority<UInt32>**

Specifies the virtual machine priority. Virtual machines are started in priority order up to the limits of a host cluster node. Valid input for this parameter is a numerical representation of the priority:

3000 = High

2000 = Medium

1000 = Low

0 - Virtual machine does not restart automatically.

Example format: -HAVMPriority 2000

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-HighlyAvailable<Boolean>**

Specifies that a virtual machine will be placed on a Hyper-V host that is part of a host cluster. Configure this setting on a virtual machine, or on a template or hardware profile that will be used to create virtual machines.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-JobGroup<Guid>**

Specifies an identifier for a series of commands that will run as a set just before the final command that includes the same job group identifier runs.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

---

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-LinuxAdministratorSSHKey<SSHKey>**

Specifies the public key file for a Linux Ssh Key.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-LinuxDomainName<String>**

Specifies a fully qualified domain name (FQDN) to be used in conjunction with Linux operating system specialization.

Example format: -LinuxDomainName "Domain01.Corp.Contoso.com"





|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-MemoryPerVirtualNumaNodeMaximumMB<Int32>**

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-MemoryWeight<Int32>**

Indicates the priority in allocating memory to a virtual machine, relative to other virtual machines on the same host. A virtual machine with a higher setting is allocated more memory resources than a virtual machine with a lower setting.

For a host running Windows Server 2008 R2 SP1 or later, 5000 = Normal, 10000 = High, 0 = Low, 1 to 10000 = Custom.

NOTE: See the examples for a specific cmdlet to determine how that cmdlet uses this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-MergeAnswerFile<Boolean>**

Specifies that the cmdlet merge the specified answer file with the specified guest operating system settings. The default value is TRUE. This parameter is used by the VMM console. You do not need to use this parameter at the command prompt.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-MonitorMaximumCount<Int32>**

Specifies the maximum number of monitors supported by a virtual video adapter.

Example format: -MonitorMaximumCount 3

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-MonitorMaximumResolution<String>**

Specifies, as a string, the value that represents the maximum possible monitor resolution of a virtual video adapter. Valid values are: "1024x768", "1280x1024", "1600x1200", "1920x1200". Default value: "1280x1024"

Example format: -MonitorResolutionMaximum "1600x1200"

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | false |
| Position? | named |

|                             |       |
|-----------------------------|-------|
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Name<String>**

Specifies the name of a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-NetworkUtilizationMbps<Int32>**

Specifies, in megabits per second (Mbps), the amount of bandwidth on the host's network that can be used by a specific virtual machine.

Example format: -NetworkUtilization 10

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-NumalsolationRequired<Boolean>**

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | false |

|                             |       |
|-----------------------------|-------|
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-NumLock<Boolean>**

Enables the BIOS value for NumLock on a virtual machine (or on a template or hardware profile that is used to create virtual machines) on a Hyper-V host. This parameter does not apply to virtual machines on VMware ESX hosts, or on Citrix XenServer hosts.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-OperatingSystem<OperatingSystem>**

Specifies the type of operating system for a virtual machine. To list the names of all available operating systems in VMM, type: "Get-SCOperatingSystem".

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-OrganizationName<String>**

Specifies the name of the organization for the person in whose name a virtual machine is registered.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Owner<String>**

Specifies the owner of a VMM object in the form of a valid domain user account.

Example format: -Owner "Contoso\ReneeLo"

Example format: -Owner "ReneeLo@Contoso"

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-ProductKey<String>**

Specifies a product key. The product key is a 25-digit number that identifies the product license. A product key can be used to register VMM or an operating system to be installed on a virtual machine or host.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-QuotaPoint<UInt32>**

Specifies a quota that limits the number of virtual machines self-service users can deploy.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RemoveCapabilityProfile**

Removes one or more specified capability profile objects.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-RemoveDomainJoinOrganizationalUnit**

Removes the organizational unit that the computer joined during setup.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RemoveServerFeatures**

Removes all server roles and features from a template.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |



---

## **-SQLProfile<SQLProfile>**

Specifies a SQL Server profile object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-Tag<String>**

Specifies a word or phrase to associate with an object so that you can search for all objects with the specified set of tags. You can search for a subset of tags, or you can search for the full set of tags.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-TimeZone<Int32>**

Specifies a number (an index) that identifies a geographical region that shares the same standard time. For a list of time zone indexes, see "Microsoft Time Zone Index Values" at: <http://go.microsoft.com/fwlink/?LinkId=120935>. If no time zone is specified, the default time zone used for a virtual machine is the same time zone setting that is on the virtual machine host.

Example format to specify the GMT Standard Time zone: -TimeZone 085

|               |       |
|---------------|-------|
| Aliases       | none  |
| Required?     | false |
| Position?     | named |
| Default Value | none  |

---

|                             |       |
|-----------------------------|-------|
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-UnattendSettings<Dictionary<String>>**

Specifies a key/value pair consisting of string, string.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-UserRole<UserRole>**

Specifies a user role object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VirtualNumaNodesPerSocketMaximum<Byte>**

|               |       |
|---------------|-------|
| Aliases       | none  |
| Required?     | false |
| Position?     | named |
| Default Value | none  |

|                             |       |
|-----------------------------|-------|
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VirtualVideoAdapterEnabled<Boolean>**

Enables, when set to \$True, the Microsoft Synthetic 3D Virtual Video Adapter for virtual machines. You can enable the Virtual Video Adapter directly on a virtual machine, or on a template or hardware profile that will be used to create virtual machines.

REQUIRED: You can enable the Microsoft Synthetic 3D Virtual Video Adapter for a virtual machine only if that virtual machine is deployed on a host running Windows Server 2008 R2 SP1 (with the Remote Desktop Services role and Remote Desktop Virtual Graphics role service installed) or later or if the virtual machine is stored in a library in a stopped state (hardware changes to a stored virtual machine can only be made if the virtual machine does not have snapshots). Enabling the Microsoft Synthetic 3D Virtual Video Adapter on a virtual machine stored in a library will limit placement of that machine to hosts running Windows Server 2008 R2 SP1 (with the Remote Desktop Services role and Remote Desktop Virtual Graphics role service installed) or later.

Example format: -VirtualVideoAdapterEnabled \$TRUE

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMTemplate<Template>**

Specifies a VMM template object used to create virtual machines.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | Template       |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

---

## **-Workgroup<String>**

Specifies on a new or existing template, on a new or existing guest operating system profile, or on a new virtual machine the name of the workgroup to which you want to join a virtual machine. You can use this parameter to override the existing value on a template or on a guest operating system profile.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **Template**

## **Notes**

- Requires a VMM virtual machine template object, which can be retrieved by using the Get-SCVMTemplate cmdlet.

## **Examples**

### **1: Specify an amount of memory for an existing virtual machine template.**

The first command gets the virtual machine template object named VMTemplate01 from the VMM library and stores the object in the \$Template variable.

The second command changes the memory value for VMTemplate01 to 1024 MB. This is the amount of memory on the host that a virtual machine created by using this template will use.

```
PS C:\> $Template = Get-SCVMTemplate | where {$_.Name -eq "VMTemplate01"}
```

```
PS C:\> Set-SCVMTemplate -Template $Template -MemoryMB 1024
```

---

## 2: Specify a new owner for multiple highly available virtual machine templates.

The first command gets all highly available virtual machine template objects and stores the objects in the \$VMTemplates object array.

The second command uses a foreach loop to specify a new owner for each of the templates in the \$VMTemplates array.

For information about the standard Windows PowerShell foreach loop statement, type: "Get-Help about\_ForEach".

```
PS C:\> $VMTemplates = Get-SCVMTemplate | where {$_.IsHighlyAvailable}
PS C:\> foreach ($VMTemplate in $VMTemplates) {Set-SCVMTemplate $VMTemplate -Owner
"Contoso\NevenSokec"}
```

## 3: Specify a new owner for all virtual machine templates that have the same owner.

This command gets all virtual machine template objects from the VMM library, selects only those objects whose owner is "Contoso\ReneeLo", and then passes each object to the Set-SCVMTemplate cmdlet which changes the owner to Contoso\NevenSokec.

```
PS C:\> Get-SCVMTemplate -VMMServer "VMMServer01.Contoso.com" | where {$_.Owner -eq
"Contoso\ReneeLo"} | Set-SCVMTemplate -Owner "Contoso\NevenSokec"
```

## 4: Enable Dynamic Memory for an existing virtual machine template.

The first command gets the virtual machine template object named VMTemplate04 from the VMM library and stores the object in the \$VMTemplate variable.

The second command enables Dynamic Memory, changes the startup memory value for VMTemplate04 to 1024 MB (this is the amount of memory on the host that a virtual machine created by using this template will be allocated upon startup) and sets the maximum memory value to 2048 MB (this is the maximum amount of memory that can be allocated to a virtual machine created by using this template). A virtual machine using Dynamic Memory can only be deployed on a host that is running Windows Server 2008 R2 SP1 or later.

```
PS C:\> $VMTemplate = Get-SCVMTemplate | where {$_.Name -eq "VMTemplate04"}
PS C:\> Set-SCVMTemplate -VMTemplate $VMTemplate -DynamicMemoryEnabled $True -MemoryMB 1024
-DynamicMemoryMaximumMB 2048
```

## 5: Specify the root password for an existing customizable Linux virtual machine template.

The first command gets the virtual machine template object named LinuxVMTemplate01 from the VMM library and stores the object in the \$Template variable.

---

The second command gets a username (root) and password via the Get-Credential command and stores the result in the \$Creds variable.

The last command sets the root account password to the password stored in the \$Creds variable.

```
PS C:\> $Template = Get-SCVMTemplate | where {$_.Name -eq "LinuxVMTemplate01"}
```

```
PS C:\> $Creds = Get-Credential
```

```
PS C:\> Set-SCVMTemplate -Template $Template -LocalAdministratorCredential $Creds
```

## Related topics

[Get-SCVMTemplate](#)

[New-SCVMTemplate](#)

[Remove-SCVMTemplate](#)

---

# Set-SCVPNConnection

---

## Set-SCVPNConnection

Sets the properties of a VPN connection.

### Syntax

Parameter Set: Default

```
Set-SCVPNConnection [-VPNConnection] <VPNConnection> [-AuthenticationMethod {PSKOnly | MachineCertificates} ] [-AuthenticationTransformConstants {SHA256128 | MD596 | SHA196 | GCMAES128 | GCMAES192 | GCMAES256} ] [-Certificate <ClientCertificate> ] [-CipherTransformConstants {DES | DES3 | AES128 | AES192 | AES256 | GCMAES128 | GCMAES192 | GCMAES256} ] [-Description <String> ] [-DHGroup {None | Group1 | Group2 | Group14 | ECP256 | ECP384 | Group24} ] [-EncryptionMethod {DES | DES3 | AES128 | AES192 | AES256} ] [-IntegrityCheckMethod {MD5 | SHA1 | SHA256 | SHA384} ] [-JobVariable <String> ] [-Name <String> ] [-PFSGroup {None | PFS1 | PFS2 | PFS2048 | ECP256 | ECP384 | PFSMM | PFS24} ] [-PROTipID <Guid> ] [-RunAsynchronously] [-Secret <RunAsAccount> ] [-TargetIPv4VPNAddress <String> ] [-TargetIPv6VPNAddress <String> ] [-VMMServer <ServerConnection> ] [ <CommonParameters> ]
```

### Detailed Description

The Set-SCVPNConnection cmdlet updates the properties of a virtual private network (VPN) connection.

### Parameters

#### **-AuthenticationMethod<VPNAuthenticationMethod>**

Specifies a virtual private network (VPN) authentication method. Valid values are: PSKOnly, MachineCertificates.

The acceptable values for this parameter are:

|                     |  |
|---------------------|--|
| PSKOnly             |  |
| MachineCertificates |  |

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

-

### **AuthenticationTransformConstants<VPNAuthenticationTransformConstants>**

Specifies a virtual private network (VPN) transform constant. Valid values are: SHA256128, MD596, SHA196, GCMAES128, GCMAES192, GCMAES256.

The acceptable values for this parameter are:

|           |  |
|-----------|--|
| SHA256128 |  |
| MD596     |  |
| SHA196    |  |
| GCMAES128 |  |
| GCMAES192 |  |
| GCMAES256 |  |

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Certificate<ClientCertificate>**

Specifies a security certificate object.



|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-CipherTransformConstants<VPNCipherTransformConstants>**

Specifies a virtual private network cipher transform constants. Valid values are: DES, DES3, AES128, AES192, AES256, GCMAES128, GCMAES192, GCMAES256.

The acceptable values for this parameter are:

|           |  |
|-----------|--|
| DES       |  |
| DES3      |  |
| AES128    |  |
| AES192    |  |
| AES256    |  |
| GCMAES128 |  |
| GCMAES192 |  |
| GCMAES256 |  |

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-Description<String>**

States a description for the specified object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-DHGroup<VPNDHGroup>**

The acceptable values for this parameter are:

|         |  |
|---------|--|
| None    |  |
| Group1  |  |
| Group2  |  |
| Group14 |  |
| ECP256  |  |
| ECP384  |  |
| Group24 |  |

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-EncryptionMethod<VPNEncryptionMethod>**

Specifies a virtual private network (VPN) encryption method. Valid values are: DES, DES3, AES128, AES192, AES256.

The acceptable values for this parameter are:

|        |  |
|--------|--|
| DES    |  |
| DES3   |  |
| AES128 |  |
| AES192 |  |
| AES256 |  |

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-IntegrityCheckMethod<VPNIntegrityCheckMethod>**

Specifies a virtual private network (VPN) integrity check method. Valid values are: MD5, SHA1, SHA256, SHA384.

The acceptable values for this parameter are:

|        |  |
|--------|--|
| MD5    |  |
| SHA1   |  |
| SHA256 |  |
| SHA384 |  |

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | false |

|                             |       |
|-----------------------------|-------|
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Name<String>**

Specifies the name of a VMM object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PFSGroup<VPNPFSGroup>**

The acceptable values for this parameter are:

|      |  |
|------|--|
| None |  |
| PFS1 |  |

|         |  |
|---------|--|
| PFS2    |  |
| PFS2048 |  |
| ECP256  |  |
| ECP384  |  |
| PFSMM   |  |
| PFS24   |  |

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

**-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

**-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | false |

|                             |       |
|-----------------------------|-------|
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Secret<RunAsAccount>**

Specifies a Run As account.

To obtain a Run As account, use the Get-SCRunAsAccount cmdlet.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-TargetIPv4VPNAddress<String>**

Specifies a target virtual private network (VPN) address in IPv4 format.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-TargetIPv6VPNAddress<String>**

Specifies a target virtual private network (VPN) address in IPv6 format.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-VPNConnection<VPNConnection>**

Specifies a virtual private network (VPN) connection object.

To obtain a VPN connection object, use the Get-SCVPNConnection cmdlet.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

---

## Related topics

[Add-SCVPNConnection](#)

[Get-SCVPNConnection](#)

[Remove-SCVPNConnection](#)



---

# Start-SCComplianceScan

---

## Start-SCComplianceScan

Starts a compliance scan of a managed computer or host cluster.

### Syntax

Parameter Set: VMHostCluster

```
Start-SCComplianceScan -VMHostCluster <HostCluster> [-Baseline <Baseline> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: VMManagedComputer

```
Start-SCComplianceScan -VMManagedComputer <VMManagedComputer> [-Baseline <Baseline> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Start-SCComplianceScan cmdlet starts a compliance scan of a managed computer or host cluster. During a compliance scan, the specified managed computer or host cluster is compared against assigned baselines, and the resulting compliance state is returned.

### Parameters

#### **-Baseline<Baseline>**

Specifies a VMM baseline object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

---

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-VMHostCluster<HostCluster>**

Specifies a VMM host cluster object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-VMMManagedComputer<VMMManagedComputer>**

Specifies a computer object that is managed by VMM.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **ComplianceStatus**
- **VMHostCluster**

## Examples

### 1: Scan a host against a given baseline.

The first command gets the host object named VMHost01 and stores the object in the \$VMHost variable.

The second command gets the compliance status object for the host stored in \$VMHost01 and stores the object in the \$Compliance variable.

The next three lines use a foreach statement to loop through the baseline compliance status objects for the host. If the baseline named Security Baseline is found, then the fifth command stores it in the \$Baseline variable.

The last command starts the compliance scan on the host, using the baseline stored in \$Baseline, which in this example is Security Baseline.

```
PS C:\> $VMHost = Get-SCVMHost -ComputerName "VMHost01"
PS C:\> $Compliance = Get-SCComplianceStatus -VMMManagedComputer $VMHost.ManagedComputer
PS C:\> foreach($bsc in $Compliance.BaselineLevelComplianceStatus)`
PS C:\> {if ($bsc.Baseline.Name -eq "Security Baseline")`
PS C:\> {$Baseline = $bsc.Baseline; break}}
PS C:\> Start-SCComplianceScan -VMMManagedComputer $VMHost.ManagedComputer -Baseline
$Baseline
```

## Related topics

[Get-SCComplianceStatus](#)

---

# Start-SCDynamicOptimization

---

## Start-SCDynamicOptimization

Starts dynamic optimization on a host cluster or host group.

### Syntax

Parameter Set: ByHostCluster

```
Start-SCDynamicOptimization [-VMHostCluster] <HostCluster> [-VMMServer <ServerConnection> ]  
[-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByHostGroup

```
Start-SCDynamicOptimization [-VMHostGroup] <HostGroup> [-VMMServer <ServerConnection> ] [-  
WhatIf] [ <CommonParameters>]
```

### Detailed Description

The Start-SCDynamicOptimization cmdlet manually starts the dynamic optimization process for a host cluster or host group.

### Parameters

#### **-VMHostCluster<HostCluster>**

Specifies a VMM host cluster object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | 1     |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-VMHostGroup<HostGroup>**

Specifies a virtual machine host group object or an array of host group objects.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | 1     |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **DynamicOptimizationResults**

## Examples

### 1: Optimize hosts in a cluster.

The first command gets the host cluster object named Cluster01 and stores the object in the \$Cluster variable.

The second command initiates the dynamic optimization process for the cluster stored in \$Cluster.

```
PS C:\> $Cluster = Get-SCVMHostCluster "Cluster01"
```

```
PS C:\> Start-SCDynamicOptimization -VMHostCluster $Cluster
```

### 2: Optimize hosts in a host group.

The first command gets the host group object named HostGroup01 and stores the object in the \$HostGroup variable.

The second command initiates the dynamic optimization process for the host group stored in \$HostGroup.

```
PS C:\> $HostGroup = Get-SCVMHostGroup "HostGroup01"
```

```
PS C:\> Start-SCDynamicOptimization -VMHostGroup $HostGroup
```

## Related topics

[Get-SCVMHostCluster](#)

---

# Start-SCService

---

## Start-SCService

Starts a VMM service and all virtual machines within the service.

### Syntax

Parameter Set: Default

```
Start-SCService [-Service] <Service> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Start-SCService cmdlet starts a Virtual Machine Manager (VMM) service and all virtual machines within the service. To stop a service, use the Stop-SCService cmdlet.

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|



|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Service<Service>**

Specifies a VMM service object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

---

|                             |                |
|-----------------------------|----------------|
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- `VirtualMachine[]`

## Examples

### 1: Start a specified service.

The first command gets the service object named Service01 and stores the object in the \$Service variable.

The second command starts the service in \$Service, which starts all of the virtual machines in the service.

```
PS C:\> $Service = Get-SCService -Name "Service01"
```

```
PS C:\> Start-SCService -Service $Service
```

## Related topics

[Get-SCService](#)

[New-SCService](#)

[Read-SCService](#)

[Remove-SCService](#)

[Resume-SCService](#)

[Set-SCService](#)

[Stop-SCService](#)

[Suspend-SCService](#)

[Update-SCService](#)

---

# Start-SCUpdateRemediation

---

## Start-SCUpdateRemediation

Initiates the action of installing one or more updates on a managed server that are required from an assigned baseline.

### Syntax

Parameter Set: ClusterNodeRemediationWithJobgroup

```
Start-SCUpdateRemediation -JobGroup <Guid> -VMHost <Host> -VMHostCluster <HostCluster> [-Baseline <Baseline> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-SuspendReboot] [-Updates <List<SoftwareUpdate>> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ClusterRemediationWithJobgroup

```
Start-SCUpdateRemediation -JobGroup <Guid> -StartNow -VMHostCluster <HostCluster> [-BypassMaintenanceModeCheck] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-UseLiveMigration] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: VMHostCluster

```
Start-SCUpdateRemediation -RemediateAllClusterNodes -VMHostCluster <HostCluster> [-Baseline <Baseline> ] [-BypassMaintenanceModeCheck] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-SuspendReboot] [-Updates <List<SoftwareUpdate>> ] [-UseLiveMigration] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: VMMManagedComputer

```
Start-SCUpdateRemediation -VMMManagedComputer <VMMManagedComputer> [-Baseline <Baseline> ] [-EnableMaintenanceMode] [-JobGroup <Guid> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-StartNow] [-SuspendReboot] [-Updates <List<SoftwareUpdate>> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

## Detailed Description

The Start-SCUpdateRemediation cmdlet initiates the action of installing one or more non-compliant updates on a managed server that are required from an assigned baseline.

## Parameters

### **-Baseline<Baseline>**

Specifies a VMM baseline object.

---

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-BypassMaintenanceModeCheck**

Indicates that cluster patching continues node by node even if a cluster node is already in maintenance mode. By default, cluster remediation fails if any of the cluster nodes are already in maintenance mode.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-EnableMaintenanceMode**

Enables maintenance mode for a stand-alone VMM management server that also serves as a host. By default, when remediating a stand-alone host, VMM does not put it into maintenance mode.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

---

## **-JobGroup<Guid>**

Specifies an identifier for a series of commands that will run as a set just before the final command that includes the same job group identifier runs.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-RemediateAllClusterNodes**

Indicates that update remediation is performed on all nodes of a cluster.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-StartNow**

Identifies the last command of a jobgroup and starts running the commands within the jobgroup. This parameter must be used with the JobGroup parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-SuspendReboot**

Indicates that the computer that is being updated will not reboot when a reboot is required by a software update.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-Updates<List<SoftwareUpdate>>**

Specifies one or more software updates.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-UseLiveMigration**

Indicates that live migration is used during a cluster remediation.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

---

## **-VMHost<Host>**

Specifies a virtual machine host object. VMM supports Hyper-V hosts, VMware ESX hosts, and Citrix XenServer hosts.

For more information about each type of host, type: "Get-Help Add-SCVMHost -detailed". See the examples for a specific cmdlet to determine how that cmdlet uses this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VMHostCluster<HostCluster>**

Specifies a VMM host cluster object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-VMMManagedComputer<VMMManagedComputer>**

Specifies a computer object that is managed by VMM.

|                        |                |
|------------------------|----------------|
| Aliases                | none           |
| Required?              | true           |
| Position?              | named          |
| Default Value          | none           |
| Accept Pipeline Input? | True (ByValue) |



---

|                             |       |
|-----------------------------|-------|
| Accept Wildcard Characters? | false |
|-----------------------------|-------|

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **ComplianceStatus**
- **VMMManagedComputer**

## **Examples**

### **1: Bring a host into compliance with a specified baseline.**

The first command gets the host object named VMHost01 and stores the object in the \$VMHost variable.

The second command gets the compliance status for host VMHost01 and stores the status object in the \$Compliance variable.

The third command gets the baseline object named Security Baseline and stores the object in the \$Baseline variable.

The fourth command starts a compliance scan on VMHost01 against the Security Baseline baseline and stores the results in the \$Compliance variable.

The last command starts remediation of VMHost01 to bring it into compliance with the Security Baseline baseline.

---

```
PS C:\> $VMHost = Get-SCVMHost -ComputerName "VMHost01"
PS C:\> $Compliance = Get-SCComplianceStatus -VMMManagedComputer $VMHost.ManagedComputer
PS C:\> $Baseline = Get-SCBaseline -Name "Security Baseline"
PS C:\> $Compliance = Start-SCComplianceScan -VMMManagedComputer $VMHost.ManagedComputer -
Baseline $Baseline
PS C:\> Start-SCUpdateRemediation -VMMManagedComputer $VMHost.ManagedComputer -Baseline
$Baseline
```

## Related topics

[Get-SCBaseline](#)

[Start-SCComplianceScan](#)

---

# Start-SCUpdateServerSynchronization

---

## Start-SCUpdateServerSynchronization

Initiates a synchronization between the VMM update server and WSUS.

### Syntax

Parameter Set: Default

```
Start-SCUpdateServerSynchronization [-UpdateServer] <UpdateServer> [-  
ForceFullUpdateCatalogImport] [-JobVariable <String> ] [-PROTipID <Guid> ] [-  
RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Start-SCUpdateServerSynchronization cmdlet initiates a synchronization between the Virtual Machine Manager (VMM) update server and Windows Server Update Services (WSUS).

The first time this synchronization runs, it imports the full WSUS update catalog into VMM. On subsequent synchronizations, an incremental import is performed by default. However, you can force a full catalog import at any time by using Start-SCUpdateServerSynchronization with the ForceFullUpdateCatalogImport parameter.

### Parameters

#### -ForceFullUpdateCatalogImport

Indicates that the VMM update server will import the full update catalog from Windows Server Update Services. On subsequent synchronizations, an incremental import of the catalog is the default import.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-UpdateServer<UpdateServer>**

Specifies a VMM update server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **UpdateServer**

---

## Examples

### 1: Start an update server synchronization.

The first command gets the update server object named WSUSComputer01 and stores the object in the \$UpdateServer variable.

The second command starts synchronization between WSUSComputer01 and Microsoft Update.

```
PS C:\> $UpdateServer = Get-SCUpdateServer -ComputerName "WSUSComputer01"
```

```
PS C:\> Start-SCUpdateServerSynchronization -UpdateServer $UpdateServer
```

### 2: Force a full import of the Microsoft Update Catalog.

The first command gets the update server object named WSUSComputer01 and stores the object in the \$UpdateServer variable.

The second command starts synchronization between WSUSComputer01 and Microsoft update, specifying that the full catalog is imported.

```
PS C:\> $UpdateServer = Get-SCUpdateServer -ComputerName "WSUSComputer01"
```

```
PS C:\> Start-SCUpdateServerSynchronization -UpdateServer $UpdateServer -  
ForceFullUpdateCatalogImport
```

## Related topics

[Get-SCUpdateServer](#)

---

# Start-SCVirtualMachine

---

## Start-SCVirtualMachine

Starts a virtual machine managed by VMM.

### Syntax

Parameter Set: SingleVM

```
Start-SCVirtualMachine [-VM] <VM> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

### Detailed Description

The Start-SCVirtualMachine cmdlet starts one or more virtual machines on hosts managed by Virtual Machine Manager (VMM) when the machines are in a stopped state. Starting a stopped virtual machine restores it to a running state and returns its object in a running state. When the virtual machine is running again, you can resume activity on that virtual machine.

If you run Start-SCVirtualMachine on a virtual machine that is already running, the cmdlet returns the object but does not change the state of the virtual machine.

To stop a running virtual machine, use the Stop-SCVirtualMachine cmdlet.

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VM<VM>**

Specifies a virtual machine object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |



---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **VirtualMachine**

## Notes

- Requires a virtual machine object, which can be retrieved by using the Get-SCVirtualMachine cmdlet.

## Examples

### 1: Start a virtual machine that is turned off.

The first command gets the virtual machine object named VM01 and stores the object in the \$VM variable.

The second command starts the virtual machine stored in \$VM, and displays information about the running virtual machine object to the user.

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM01"
```

```
PS C:\> Start-SCVirtualMachine -VM $VM
```

### 2: Start all virtual machines that are turned off.

The first command gets all virtual machine objects on VMMServer01 that are in a stopped state, and stores the objects in the \$VMs object array.

The second command passes each virtual machine object stored in \$VMs to the Start-SCVirtualMachine cmdlet, which starts each virtual machine in the array.

```
PS C:\> $VMs = Get-SCVirtualMachine -VMMServer "VMMServer01.Contoso.com" | where { $_.Status -eq "PowerOff" }
```

```
PS C:\> $VMs | Start-SCVirtualMachine
```

## Related topics

[Get-SCVirtualMachine](#)

[Move-SCVirtualMachine](#)

[New-SCVirtualMachine](#)

[Read-SCVirtualMachine](#)

---

[Register-SCVirtualMachine](#)

[Remove-SCVirtualMachine](#)

[Repair-SCVirtualMachine](#)

[Reset-SCVirtualMachine](#)

[Resume-SCVirtualMachine](#)

[Set-SCVirtualMachine](#)

[Stop-SCVirtualMachine](#)

[Suspend-SCVirtualMachine](#)

---

# Start-SCVMHost

---

## Start-SCVMHost

Starts a stopped host managed by VMM by using out-of-band management.

### Syntax

Parameter Set: Default

```
Start-SCVMHost [-VMHost] <Host> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

### Detailed Description

The Start-SCVMHost cmdlet starts a stopped host managed by Virtual Machine Manager by using out-of-band management. Prior to using this cmdlet, a host must be configured for out-of-band (OOB) management. For information about configuring the Baseboard Management Controller (BMC) settings for a host, type: "Get-Help Set-SCVMHost -detailed".

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMHost<Host>**

Specifies a virtual machine host object. VMM supports Hyper-V hosts, VMware ESX hosts, and Citrix XenServer hosts.

For more information about each type of host, type: "Get-Help Add-SCVMHost -detailed". See the examples for a specific cmdlet to determine how that cmdlet uses this parameter.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- Host
- Host[]

## Examples

### 1: Power on a specified host.

This command gets the host object named NewHost01 and starts the host using out-of-band management.

```
PS C:\> Get-SCVMHost -ComputerName "NewHost01" | Start-SCVMHost
```

## Related topics

[Add-SCVMHost](#)

[Get-SCVMHost](#)

[Restart-SCVMHost](#)

[Set-SCVMHost](#)

[Stop-SCVMHost](#)

---

# Stop-SCJob

---

## Stop-SCJob

Stops running VMM jobs.

### Syntax

Parameter Set: Default

```
Stop-SCJob [-Job] <Task> [ <CommonParameters>]
```

### Detailed Description

The Stop-SCJob cmdlet stops one or more Virtual Machine Manager (VMM) jobs that are running, and returns the object for each job in a stopped state. If the VMM job is not currently running, this cmdlet has no effect.

### Parameters

#### **-Job<Task>**

Specifies a VMM job object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

#### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

---

## Outputs

The output type is the type of the objects that the cmdlet emits.

- Task

## Notes

- Requires a VMM job object, which can be retrieved by using the Get-SCJob cmdlet.

## Examples

### 1: Stop all currently running jobs.

The first command gets all VMM job objects, passes each job object to the "where" filter to select only the jobs that are currently running, and stores the objects in the \$Job object array.

The second command passes each object in \$Job to the Stop-SCJob cmdlet, which stops each running job.

```
PS C:\> $Job = Get-SCJob | where { $_.Status -eq "Running" }
```

```
PS C:\> $Job | Stop-SCJob
```

### 2: Stop a specific running job asynchronously.

The first command gets all VMM job objects and, from the results, selects only the job on VM01 identified by job ID cb3a0f0a-9fbc-4bd0-a999-3fae8cd77177, and then stores this object in the \$Job variable.

The second command stops the job and returns the stopped job object to the user.

```
PS C:\> $Job = Get-SCJob | where { $_.ResultName -eq "VM01" -and $_.ID -eq "cb3a0f0a-9fbc-4bd0-a999-3fae8cd77177" }
```

```
PS C:\> Stop-SCJob -Job $Job
```

## Related topics

[Find-SCJob](#)

[Get-SCJob](#)

[Restart-SCJob](#)

---

# Stop-SCService

---

## Stop-SCService

Stops a VMM service and all virtual machines within the service.

### Syntax

Parameter Set: Default

```
Stop-SCService [-Service] <Service> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: DiscardSavedState

```
Stop-SCService [-Service] <Service> -DiscardSavedState [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: SaveState

```
Stop-SCService [-Service] <Service> -SaveState [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: Shutdown

```
Stop-SCService [-Service] <Service> -Shutdown [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Stop-SCService cmdlet stops a Virtual Machine Manager (VMM) service and all virtual machines within the service. To start a service, use the Start-SCService cmdlet.

### Parameters

#### -DiscardSavedState

Deletes the saved state associated with a virtual machine or service.

|               |       |
|---------------|-------|
| Aliases       | none  |
| Required?     | true  |
| Position?     | named |
| Default Value | none  |



---

|                             |       |
|-----------------------------|-------|
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|               |       |
|---------------|-------|
| Aliases       | none  |
| Required?     | false |
| Position?     | named |
| Default Value | none  |

|                             |       |
|-----------------------------|-------|
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-SaveState**

Saves the state of a virtual machine or service.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-Service<Service>**

Specifies a VMM service object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-Shutdown**

Indicates that a virtual machine, service, or a source server should be shut down. In the case of a virtual machine or service, the associated cmdlet attempts to use the operating system to shut the virtual machine down gracefully. In the case of a successful physical-to-virtual machine (P2V) conversion, the cmdlet shuts down the source server.

|           |      |
|-----------|------|
| Aliases   | none |
| Required? | true |

|                             |       |
|-----------------------------|-------|
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **VirtualMachine[]**

## **Examples**

### **1: Stop a specific service.**

The first command gets the service object named Service01 and stores the object in the \$Service variable.

The second command stops the service in \$Service, which stops all of the virtual machines in the service.

```
PS C:\> $Service = Get-SCService -Name "Service01"
```

```
PS C:\> Stop-SCService -Service $Service
```

---

## Related topics

[Get-SCService](#)

[New-SCService](#)

[Read-SCService](#)

[Remove-SCService](#)

[Resume-SCService](#)

[Set-SCService](#)

[Start-SCService](#)

[Suspend-SCService](#)

[Update-SCService](#)

---

# Stop-SCVirtualMachine

---

## Stop-SCVirtualMachine

Stops virtual machines managed by VMM.

### Syntax

Parameter Set: SingleVM

```
Stop-SCVirtualMachine [-VM] <VM> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-Shutdown] [ <CommonParameters>]
```

Parameter Set: CorrespondsToVMM2008DiscardSavedStateVM

```
Stop-SCVirtualMachine [-VM] <VM> -DiscardSavedState [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

Parameter Set: CorrespondsToVMM2008SaveStateVM

```
Stop-SCVirtualMachine [-VM] <VM> -SaveState [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

Parameter Set: CorrespondsToVMM2008StopVM

```
Stop-SCVirtualMachine [-VM] <VM> -Force [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

### Detailed Description

The Stop-SCVirtualMachine cmdlet stops one or more running virtual machines on hosts managed by Virtual Machine Manager (VMM) and returns the virtual machine object in a stopped state.

Stop-SCVirtualMachine stops a virtual machine in the same manner as shutting down the operating system on a computer. Stop-SCVirtualMachine with the Force parameter stops a virtual machine in the same manner as turning off a computer.

To resume running a stopped virtual machine, use the Start-SCVirtualMachine cmdlet.

### Parameters

#### -DiscardSavedState

Deletes the saved state associated with a virtual machine or service.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

|                             |       |
|-----------------------------|-------|
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-Force**

Forces the operation to complete.

For example:

- Remove-SCSCVMHost -Force

Forces the removal of a host object from the VMM database.

- Stop-SCVirtualMachine -Force

Stops a virtual machine.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-SaveState**

Saves the state of a virtual machine or service.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-Shutdown**

Indicates that a virtual machine, service, or a source server should be shut down. In the case of a virtual machine or service, the associated cmdlet attempts to use the operating system to shut the virtual machine down gracefully. In the case of a successful physical-to-virtual machine (P2V) conversion, the cmdlet shuts down the source server.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VM<VM>**

Specifies a virtual machine object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **VirtualMachine**



---

## Notes

- Requires a virtual machine object, which can be retrieved by using the Get-SCVirtualMachine cmdlet.

## Examples

### 1: Stop a specified virtual machine.

The first command gets the virtual machine object named VM01 and stores the object in the \$VM variable. This example assumes that only one virtual machine named VM01 exists and that it is currently in a running state.

The second command stops the virtual machine stored in \$VM (in this case, VM01) and displays information about the stopped object to the user.

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM01"
PS C:\> Stop-SCVirtualMachine -VM $VM
```

### 2: Stop multiple virtual machines.

The first command gets all virtual machine objects whose name contains the string VMM and whose current status is "Running". The command then stores those objects in the \$VMs variable.

The second command passes each virtual machine object stored in \$VMs to Stop-VM, which stops each virtual machine and displays information about the stopped virtual machines to the user.

```
PS C:\> $VMs = Get-SCVirtualMachine | where { $_.Name -match "VM" -and $_.Status -eq "Running" }
PS C:\> $VMs | Stop-SCVirtualMachine
```

## Related topics

[Get-SCVirtualMachine](#)

[Move-SCVirtualMachine](#)

[New-SCVirtualMachine](#)

[Read-SCVirtualMachine](#)

[Register-SCVirtualMachine](#)

[Remove-SCVirtualMachine](#)

[Repair-SCVirtualMachine](#)

[Reset-SCVirtualMachine](#)

[Resume-SCVirtualMachine](#)

[Set-SCVirtualMachine](#)

[Start-SCVirtualMachine](#)

[Stop-SCVirtualMachine](#)

---

[Suspend-SCVirtualMachine](#)

---

# Stop-SCVMHost

---

## Stop-SCVMHost

Stops a host managed by VMM.

### Syntax

Parameter Set: Default

```
Stop-SCVMHost [-VMHost] <Host> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-UseOutOfBandChannel] [ <CommonParameters>]
```

### Detailed Description

The Stop-SCVMHost cmdlet stops a host managed by Virtual Machine Manager (VMM) by issuing a "Power Off" command.

There are two methods by which you can stop a host: in-band and out-of-band. The default parameter set uses the in-band method to shut down the host. To use the out-of-band method, the host must be configured for out-of-band (OOB) management. For information about configuring the Baseboard Management Controller (BMC) settings for a host, type: "Get-Help Set-SCVMHost -detailed".

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-UseOutOfBandChannel**

Indicates that the out-of-band management channel should be used to perform the operation. This parameter should be used only when the computer is not responsive.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-VMHost<Host>**

Specifies a virtual machine host object. VMM supports Hyper-V hosts, VMware ESX hosts, and Citrix XenServer hosts.

For more information about each type of host, type: "Get-Help Add-SCVMHost -detailed". See the examples for a specific cmdlet to determine how that cmdlet uses this parameter.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **Host**
- **Host[]**

## **Examples**

### **1: Power off a specified host using in-band management**

This command gets the host object named NewHost01 and powers off the host. The command prompts you to confirm the action before proceeding.

```
PS C:\> Get-SCVMHost -ComputerName "NewHost01" | Stop-SCVMHost -Confirm
```

### **2: Power off a specified host using out-of-band management.**

This command gets the host object named NewHost01 and powers off the host using out-of-band management. The command prompts you for confirmation before proceeding.

```
PS C:\> Get-SCVMHost -ComputerName "NewHost01" | Stop-SCVMHost -UseOutOfBandChannel -Confirm
```

---

## Related topics

[Add-SCVMHost](#)

[Get-SCVMHost](#)

[Restart-SCVMHost](#)

[Set-SCVMHost](#)

[Start-SCVMHost](#)

---

# Suspend-SCService

---

## Suspend-SCService

Pauses a VMM service and all virtual machines within the service.

### Syntax

Parameter Set: Default

```
Suspend-SCService [-Service] <Service> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Suspend-SCService pauses a Virtual Machine Manager (VMM) service and all virtual machines within the service. To resume the service, use the Resume-SCService cmdlet.

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Service<Service>**

Specifies a VMM service object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|



---

|                             |                |
|-----------------------------|----------------|
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- `VirtualMachine[]`

## Examples

### 1: Suspend a specific service.

The first command gets the service object named Service01 and stores the object in the \$Service variable.

The second command suspends the service in \$Service, which pauses all of the virtual machines in the service.

```
PS C:\> $Service = Get-SCService -Name "Service01"
```

```
PS C:\> Suspend-SCService -Service $Service
```

## Related topics

[Get-SCService](#)

[New-SCService](#)

[Read-SCService](#)

[Remove-SCService](#)

[Resume-SCService](#)

[Set-SCService](#)

[Start-SCService](#)

[Stop-SCService](#)

[Update-SCService](#)

---

# Suspend-SCVirtualMachine

---

## Suspend-SCVirtualMachine

Suspends a virtual machine managed by VMM.

### Syntax

Parameter Set: Default

```
Suspend-SCVirtualMachine [-VM] <VM> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

### Detailed Description

The Suspend-SCVirtualMachine cmdlet suspends one or more virtual machines deployed on hosts managed by Virtual Machine Manager (VMM). Suspending a virtual machine pauses activity on that virtual machine and returns its object in a paused state.

To resume running a suspended virtual machine, use the Resume-SCVirtualMachine cmdlet.

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VM<VM>**

Specifies a virtual machine object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

---

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **VirtualMachine**

## Notes

- Requires a virtual machine object, which can be retrieved by using the `Get-SCVirtualMachine` cmdlet.

## Examples

### 1: Suspend a specified virtual machine.

The first command gets the virtual machine object named VM01 and stores the object in the `$VM` variable.

The second command pauses the virtual machine stored in `$VM` (in this case, VM01) and displays information about the paused object to the user.

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM01"
```

```
PS C:\> Suspend-SCVirtualMachine -VM $VM
```

## Related topics

[Get-SCVirtualMachine](#)

[Move-SCVirtualMachine](#)

[New-SCVirtualMachine](#)

[Read-SCVirtualMachine](#)

[Register-SCVirtualMachine](#)

[Remove-SCVirtualMachine](#)

[Repair-SCVirtualMachine](#)

[Reset-SCVirtualMachine](#)

[Resume-SCVirtualMachine](#)

[Set-SCVirtualMachine](#)

[Start-SCVirtualMachine](#)

[Stop-SCVirtualMachine](#)

---

# Test-SCCapabilityProfile

---

## Test-SCCapabilityProfile

Validates the settings of a capability profile against a virtual machine, hardware profile, or virtual machine template.

### Syntax

Parameter Set: HardwareProfile

```
Test-SCCapabilityProfile -HardwareProfile <HardwareProfile> [-CapabilityProfile <CapabilityProfile> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: Template

```
Test-SCCapabilityProfile -VMTemplate <Template> [-CapabilityProfile <CapabilityProfile> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: VM

```
Test-SCCapabilityProfile -VM <VM> [-CapabilityProfile <CapabilityProfile> ] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Test-SCCapabilityProfile cmdlet validates the settings of a capability profile against the virtual machine, hardware profile, or virtual machine template to which the profile is attached.

### Parameters

#### **-CapabilityProfile<CapabilityProfile>**

Specifies a capability profile object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-HardwareProfile<HardwareProfile>**

Specifies a hardware profile object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-VM<VM>**

Specifies a virtual machine object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

---

## **-VMTemplate<Template>**

Specifies a VMM template object used to create virtual machines.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Examples**

### **1: Validate a capability profile for a virtual machine in a private cloud.**

The first command gets the virtual machine object named CloudVM01 and stores the object in the \$VM variable.

The second command tests the settings of the capability profile attached to the virtual machine stored in \$VM and stores the results in the \$ProfileTest variable.

The last command displays any validation errors that occurred during testing.

```
PS C:\> $VM = Get-SCVirtualMachine -Name "CloudVM01"
```

```
PS C:\> $ProfileTest = Test-SCCapabilityProfile -VM $VM
```

```
PS C:\> $ProfileTest.ValidationErrors
```

## **Related topics**

[Get-SCCapabilityProfile](#)

[New-SCCapabilityProfile](#)

[Remove-SCCapabilityProfile](#)

[Set-SCCapabilityProfile](#)

---

# Test-SCDomainCredential

---

## Test-SCDomainCredential

Tests a credential or user name to verify that it authenticates in the domain.

### Syntax

Parameter Set: Credential

```
Test-SCDomainCredential [-Credential] <PSCredential> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: UserName

```
Test-SCDomainCredential [-UserName] <String> [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Test-SCDomainCredential cmdlet tests a credential object or user name to verify that it authenticates in the domain.

### Parameters

#### **-Credential<PSCredential>**

Specifies a credential object or, for some cmdlets, a Run As account object that contains the user name and password of an account that has permission to perform this action. Or, in the case of Restart-SCJob, has permission to complete a restarted task.

For more information about the PSCredential object, type: "Get-Help Get-Credential".

For more information about Run As accounts, type: "Get-Help New-SCRunAsAccount".

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |



---

## **-UserName<String>**

Specifies a the name of a user. Enter a user name with the format Domain\User.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **Boolean**

---

## Examples

### 1: Test the validity of a credential object.

The first command prompts you for a username and password, creates a PSCredential object, and stores the object in the \$Creds variable.

The second command validates the credential object in \$Creds and returns either True or False.

```
PS C:\> $Creds = Get-Credential
```

```
PS C:\> Test-SCDomainCredential -Credential $Creds
```

### 2: Test the validity of a user name.

This command tests the validity of the user name ReneeLo and returns either True or False.

```
PS C:\> Test-SCDomainCredential -UserName "ReneeLo"
```

---

# Test-SCLoadBalancer

---

## Test-SCLoadBalancer

Tests a load balancer.

### Syntax

Parameter Set: Default

```
Test-SCLoadBalancer [-LoadBalancerAddress] <String> -ConfigurationProvider  
<ConfigurationProvider> -Port <UInt16> -RunAsAccount <RunAsAccount> [-JobVariable <String> ]  
[-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [  
<CommonParameters>]
```

### Detailed Description

The Test-SCLoadBalancer cmdlet runs tests against a load balancer and returns the results to the user.

### Parameters

#### **-ConfigurationProvider<ConfigurationProvider>**

Specifies a configuration provider object. A configuration provider is a plug-in to VMM that translates VMM PowerShell commands to API calls that are specific to a type of load balancer. If no configuration provider is specified, VMM uses the Manufacturer and Model information to choose an available configuration provider. If no configuration provider is found, the load balancer will not be added.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-LoadBalancerAddress<String>**

Specifies the fully qualified domain name (FQDN) or IP address of a load balancer. Usual formats are FQDN, IPv4 or IPv6 addresses, but check with the load balancer manufacturer for the valid format for your load balancer.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-Port<UInt16>**

Specifies the network port to use when adding an object or creating a connection. Valid values are: 1 to 4095.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RunAsAccount<RunAsAccount>**

Specifies a Run As account that contains credentials with permission to perform this action.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **LoadBalancer**

## **Examples**

### **1: Test a specified load balancer.**

The first command gets the configuration provider object for the manufacturer LBManufacturer and model LB01, and stores the object in the \$ConfigProvider variable.

The second command gets the Run As account named LBRunAsAccount and stores the object in the \$RunAsAccount variable.

The last command tests the load balancer with the address LB01.Contoso.com, using providing the Run As account stored in \$RunAsAccount as credentials to run the tests. The command then displays the results to the user.

```
PS C:\> $ConfigProvider = Get-SCConfigurationProvider | where { $_.Type -eq "LoadBalancer" -and $_.Manufacturer -eq "LBManufacturer" -and $_.Model -eq "LB01" }
```

```
PS C:\> $RunAsAccount = Get-SCRunAsAccount -Name "LBRunAsAcct"
```

```
PS C:\> Test-SCLoadBalancer -LoadBalancerAddress "LB01.Contoso.com" -Port 123 -ConfigurationProvider $ConfigProvider -RunAsAccount $RunAsAccount
```

---

## Related topics

[Add-SCLoadBalancer](#)

[Get-SCLoadBalancer](#)

[Read-SCLoadBalancer](#)

[Remove-SCLoadBalancer](#)

[Set-SCLoadBalancer](#)

---

# Test-SCNetworkGateway

---

## Test-SCNetworkGateway

Tests a network gateway.

### Syntax

Parameter Set: TestGateway

```
Test-SCNetworkGateway -ConfigurationProvider <ConfigurationProvider> -ConnectionString  
<String> -RunAsAccount <RunAsAccount> [-JobVariable <String> ] [-PROTipID <Guid> ] [-  
RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: TestPromiscuous

```
Test-SCNetworkGateway -ConfigurationProvider <ConfigurationProvider> [-JobVariable <String>  
] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [  
<CommonParameters>]
```

### Detailed Description

The Test-SCNetworkGateway cmdlet tests a network gateway in Virtual Machine Manager (VMM).

### Parameters

#### **-ConfigurationProvider<ConfigurationProvider>**

Specifies a configuration provider object. A configuration provider is a plug-in to VMM that translates VMM PowerShell commands to API calls that are specific to a type of load balancer. If no configuration provider is specified, VMM uses the Manufacturer and Model information to choose an available configuration provider. If no configuration provider is found, the load balancer will not be added.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |



---

## **-ConnectionString<String>**

Specifies the information required to connect to the virtual switch extension manager.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-RunAsAccount<RunAsAccount>**

Specifies a Run As account that contains credentials with permission to perform this action.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

### Related topics

[Add-SCNetworkGateway](#)

[Get-SCNetworkGateway](#)

[Read-SCNetworkGateway](#)

[Remove-SCNetworkGateway](#)

[Set-SCNetworkGateway](#)

---

# Test-SCPROTip

---

## Test-SCPROTip

Tests PRO integration between VMM and Operations Manager.

### Syntax

Parameter Set: Default

```
Test-SCPROTip [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Test-SCPROTip cmdlet tests integration between Virtual Machine Manager (VMM) and Operations Manager. After creating a connection with Operations Manager and configuring Performance and Resource Optimization (PRO) monitors, run Test-SCPROTip to validate that PRO integration is functioning correctly. Test-SCPROTip creates a new warning PRO alert in Operations Manager, invokes a remediation, and then implements the fix for the PRO tip.

### Parameters

#### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

#### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

---

## Examples

### 1. Test PRO integration between VMM and Operations Manager.

The first command runs the Test-SCPROTip operation and stores the associated job in the variable called PROTipJob.

The second command verifies the status of the test PRO Tip. The status may also be viewed in the Virtual Machine Manager job interface.

```
PS C:\> $PROTipJob = Test-SCPROTip
```

```
PS C:\> $PROTipJob
```

### Related topics

[Clear-SCPROTip](#)

[Get-SCPROTip](#)

[Invoke-SCPROTip](#)

[Set-SCPROTip](#)

---

# Test-SCServiceTemplate

---

## Test-SCServiceTemplate

Validates a service template and stores any errors in the ValidationErrors property of the service template.

### Syntax

Parameter Set: Default

```
Test-SCServiceTemplate [-ServiceTemplate] <ServiceTemplate> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-Update] [ <CommonParameters>]
```

### Detailed Description

The Test-SCServiceTemplate cmdlet validates a service template and stores the errors in the ValidationErrors property of the service template.

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-ServiceTemplate<ServiceTemplate>**

Specifies a service template object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-Update**

Updates the settings for an object.

---

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **ServiceTemplate**

## Examples

### 1: Validate a service template.

The first command gets the Beta release of the service template object named ServiceTemplate01 and stores the object in the \$SvcTemplate variable.

The second command validates the service template in \$SvcTemplate.

The third command displays the first validation error for service template from the validation error array.

```
PS C:\> $SvcTemplate = Get-SCServiceTemplate -Name "ServiceTemplate01" | where { $_.Release -eq "Beta" }
```

```
PS C:\> $UpdatedSvcTemplate = Test-SCServiceTemplate -ServiceTemplate $SvcTemplate
```

```
PS C:\> $UpdatedSvcTemplate.ValidationErrors[0]
```

## Related topics

[Get-SCServiceTemplate](#)

[New-SCServiceTemplate](#)

[Read-SCServiceTemplate](#)

[Remove-SCServiceTemplate](#)

[Resolve-SCServiceTemplate](#)

[Set-SCServiceTemplate](#)



---

# Test-SCVirtualDiskDrive

---

## Test-SCVirtualDiskDrive

Tests an existing virtual hard disk that is attached to a virtual disk drive object.

### Syntax

Parameter Set: Default

```
Test-SCVirtualDiskDrive [-VirtualDiskDrive] <VirtualDiskDrive> [-JobGroup <Guid> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

### Detailed Description

The Test-SCVirtualDiskDrive cmdlet tests an existing virtual hard disk that is attached to a virtual disk drive object.

To test a virtual hard disk, the virtual machine on which the virtual hard disk is configured must be in a stopped state.

### Parameters

#### **-JobGroup<Guid>**

Specifies an identifier for a series of commands that will run as a set just before the final command that includes the same job group identifier runs.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

---

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-VirtualDiskDrive<VirtualDiskDrive>**

Specifies a virtual disk drive object. You can attach either a virtual hard disk (for a virtual machine on any host) or a pass-through disk (for a virtual machine on a Hyper-V host or an ESX host) to a virtual disk drive object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

### **Examples**

#### **1: Test a dynamic VHD attached to a fixed-format virtual disk drive on a virtual machine.**

The first command gets the virtual machine object named VM03 and stores the object in the \$VM variable.

The second command gets the virtual disk drive object that is attached to virtual machine VM03 and stores the object in the \$VirtDiskDrive variable. This example assumes that the virtual machine has only one virtual disk drive object and that the virtual hard disk attached to the virtual disk drive is a dynamic virtual hard disk.

The third command determines whether VM03 is running. If VM03 is running, the command shuts down the virtual machine.

The last command tests the virtual hard disk stored in \$VirtDiskDrive.

```
PS C:\> $VM = Get-SCVirtualMachine -Name "VM03"
PS C:\> $VirtDiskDrive = Get-SCVirtualDiskDrive -VM $VM
PS C:\> If ($VM.Status -eq "Running") {Stop-SCVirtualMachine -VM $VM -Shutdown}
PS C:\> Test-SCVirtualDiskDrive -VirtualDiskDrive $VirtDiskDrive
```

---

## Related topics

[Compress-SCVirtualDiskDrive](#)

[Convert-SCVirtualDiskDrive](#)

[Expand-SCVirtualDiskDrive](#)

[Get-SCVirtualDiskDrive](#)

[New-SCVirtualDiskDrive](#)

[Remove-SCVirtualDiskDrive](#)

[Set-SCVirtualDiskDrive](#)

---

# Test-SCVirtualSwitchExtensionManager

---

## Test-SCVirtualSwitchExtensionManager

Tests a virtual switch extension manager.

### Syntax

Parameter Set: Default

```
Test-SCVirtualSwitchExtensionManager [-VirtualSwitchExtensionManagerConnectionString]
<String> -ConfigurationProvider <ConfigurationProvider> -RunAsAccount <RunAsAccount> [-
JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer
<ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Test-SCVirtualSwitchExtensionManager tests a virtual switch extension manager.

### Parameters

#### **-ConfigurationProvider<ConfigurationProvider>**

Specifies a configuration provider object. A configuration provider is a plug-in to VMM that translates VMM PowerShell commands to API calls that are specific to a type of load balancer. If no configuration provider is specified, VMM uses the Manufacturer and Model information to choose an available configuration provider. If no configuration provider is found, the load balancer will not be added.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RunAsAccount<RunAsAccount>**

Specifies a Run As account that contains credentials with permission to perform this action.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VirtualSwitchExtensionManagerConnectionString<String>**

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Related topics

[Add-SCVirtualSwitchExtensionManager](#)

[Get-SCVirtualSwitchExtensionManager](#)

[Read-SCVirtualSwitchExtensionManager](#)

[Remove-SCVirtualSwitchExtensionManager](#)

[Set-SCVirtualSwitchExtensionManager](#)



---

# Test-SCVMHostCluster

---

## Test-SCVMHostCluster

Validates whether hosts that are managed by VMM are suitable as a nodes of a failover cluster.

### Syntax

Parameter Set: ValidateCluster

```
Test-SCVMHostCluster -VMHostCluster <HostCluster> [-Credential <PSCredential> ] [-  
JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer  
<ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: ValidateNodes

```
Test-SCVMHostCluster -VMHost <Host[]> [-Credential <PSCredential> ] [-JobVariable <String> ]  
[-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [  
<CommonParameters>]
```

### Detailed Description

The Test-SCVMHostCluster cmdlet validates whether one or more hosts that are managed by Virtual Machine Manager (VMM) is suitable as a node of a failover cluster.

### Parameters

#### **-Credential<PSCredential>**

Specifies a credential object or, for some cmdlets, a Run As account object that contains the user name and password of an account that has permission to perform this action. Or, in the case of Restart-SCJob, has permission to complete a restarted task.

For more information about the PSCredential object, type: "Get-Help Get-Credential".

For more information about Run As accounts, type: "Get-Help New-SCRunAsAccount".

|                        |       |
|------------------------|-------|
| Aliases                | none  |
| Required?              | false |
| Position?              | named |
| Default Value          | none  |
| Accept Pipeline Input? | false |

---

|                             |       |
|-----------------------------|-------|
| Accept Wildcard Characters? | false |
|-----------------------------|-------|

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                        |       |
|------------------------|-------|
| Aliases                | none  |
| Required?              | false |
| Position?              | named |
| Default Value          | none  |
| Accept Pipeline Input? | false |

---

|                             |       |
|-----------------------------|-------|
| Accept Wildcard Characters? | false |
|-----------------------------|-------|

### **-VMHost<Host[]>**

Specifies an array of virtual machine host objects.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-VMHostCluster<HostCluster>**

Specifies a VMM host cluster object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                        |                |
|------------------------|----------------|
| Aliases                | none           |
| Required?              | false          |
| Position?              | named          |
| Default Value          | none           |
| Accept Pipeline Input? | True (ByValue) |

---

|                             |       |
|-----------------------------|-------|
| Accept Wildcard Characters? | false |
|-----------------------------|-------|

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **ClusterValidationResult**

## Examples

### 1: Validate nodes to be clustered

The first command gets the host group object named New York in All Hosts and stores the object in the \$HostGroup variable.

The second command gets all host objects with names beginning with "Cluster" from the New York host group and stores the objects in the \$Nodes variable.

The third command validates the host objects stored in \$Nodes for failover cluster creation and stores the results in \$Result.

The fourth command displays the validation result to the user.

The last command displays the location of the validation report file to the user.

```
PS C:\> $HostGroup = Get-SCVMHostGroup -Name "New York"
```

```
PS C:\> $Nodes = Get-SCVMHost | where {$_.Name -like "Cluster*" -and $_.VMHostGroup -eq $HostGroup}
```

```
PS C:\> $Result = Test-SCVMHostCluster -VMHost $Nodes
```

```
PS C:\> Write-Output $Result.ValidationResult
```

```
PS C:\> Write-Output $Result.ResultFileLocation
```

### 2: Validate an existing cluster.

The first command gets the cluster object named Cluster01 and stores the object in the \$Cluster variable.

The second command tests the cluster stored in \$Cluster and stores the results of the test in \$Result.

The third command displays the results for the user.

The last command displays the location of the validation report file for the user.

```
PS C:\> $Cluster = Get-SCVMHostCluster -Name "Cluster01"
```

```
PS C:\> $Result = Test-SCVMHostCluster -VMHostCluster $Cluster
```

```
PS C:\> Write-Output $Result.ValidationResult
```

---

PS C:\> Write-Output \$result.ResultFileLocation

## Related topics

[Get-SCVMHost](#)

[Install-SCVMHostCluster](#)

[Uninstall-SCVMHostCluster](#)

---

# Uninstall-SCVMHostCluster

---

## Uninstall-SCVMHostCluster

Uninstalls (destroys) a failover cluster managed by VMM.

### Syntax

Parameter Set: DestroyCluster

```
Uninstall-SCVMHostCluster -VMHostCluster <HostCluster> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: RemoveNodes

```
Uninstall-SCVMHostCluster -VMHost <Host[]> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Uninstall-SCVMHostCluster cmdlet unclusters the nodes of a Hyper-V host cluster managed by Virtual Machine Manager (VMM). After the cluster has been uninstalled, each host is managed by VMM as a stand-alone host.

Uninstall-SCVMHostCluster can also remove a node from a cluster when used with the VMHost parameter.

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VMHost<Host[]>**

Specifies an array of virtual machine host objects.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

---

## **-VMHostCluster<HostCluster>**

Specifies a VMM host cluster object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## **Examples**

### **1: Destroy an existing cluster.**

The first command gets the cluster object named Cluster01 and stores the object in the \$Cluster variable.

The second command destroys the cluster stored in \$Cluster (in this case, Cluster01). After the cluster is destroyed, each host will be managed by VMM as a stand-alone host.

```
PS C:\> $Cluster = Get-SCVMHostCluster -Name "Cluster01"
```

```
PS C:\> Uninstall-SCVMHostCluster -VMHostCluster $Cluster
```



---

## **Related topics**

[Add-SCVMHostCluster](#)

[Get-SCVMHostCluster](#)

[Install-SCVMHostCluster](#)

[Move-SCVMHostCluster](#)

[Read-SCVMHostCluster](#)

[Remove-SCVMHostCluster](#)

[Set-SCVMHostCluster](#)

[Test-SCVMHostCluster](#)

---

# Unregister-SCStorageFileShare

---

## Unregister-SCStorageFileShare

Unregisters a storage file share from a VM host, cluster, or library server.

### Syntax

Parameter Set: UnregisterFromHost

```
Unregister-SCStorageFileShare [-StorageFileShare] <StorageFileShare> -VMHost <Host> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RemovePermissionsFromShare] [-RunAsynchronously] [ <CommonParameters>]
```

Parameter Set: UnregisterFromCluster

```
Unregister-SCStorageFileShare [-StorageFileShare] <StorageFileShare> -VMHostCluster <HostCluster> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RemovePermissionsFromShare] [-RunAsynchronously] [ <CommonParameters>]
```

Parameter Set: UnregisterFromLibrary

```
Unregister-SCStorageFileShare [-StorageFileShare] <StorageFileShare> -LibraryServer <LibraryServer> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RemovePermissionsFromShare] [-RunAsynchronously] [ <CommonParameters>]
```

### Detailed Description

The Unregister-SCStorageFileShare cmdlet unregisters a storage file share from a virtual machine host, cluster, or library server.

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                        |       |
|------------------------|-------|
| Aliases                | none  |
| Required?              | false |
| Position?              | named |
| Default Value          | none  |
| Accept Pipeline Input? | false |

---

|                             |       |
|-----------------------------|-------|
| Accept Wildcard Characters? | false |
|-----------------------------|-------|

### **-LibraryServer<LibraryServer>**

Specifies a VMM library server object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RemovePermissionsFromShare**

Indicates that permissions are removed from the file share.

|                        |       |
|------------------------|-------|
| Aliases                | none  |
| Required?              | false |
| Position?              | named |
| Default Value          | none  |
| Accept Pipeline Input? | false |

|                             |       |
|-----------------------------|-------|
| Accept Wildcard Characters? | false |
|-----------------------------|-------|

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-StorageFileShare<StorageFileShare>**

Specifies a storage file share.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-VMHost<Host>**

Specifies a virtual machine host object. VMM supports Hyper-V hosts, VMware ESX hosts, and Citrix XenServer hosts.

For more information about each type of host, type: "Get-Help Add-SCVMHost -detailed". See the examples for a specific cmdlet to determine how that cmdlet uses this parameter.

|           |       |
|-----------|-------|
| Aliases   | none  |
| Required? | true  |
| Position? | named |

|                             |       |
|-----------------------------|-------|
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VMHostCluster<HostCluster>**

Specifies a VMM host cluster object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Examples**

### **1: Unregister a storage file share from a library server.**

The first command gets the storage file share object named FileShare01 and stores the object in the \$FileShare variable.

The second command gets the VMM library server object named LibraryServer01 and stores the object in the \$LibServer variable.

The last command unregisters FileShare01 from LibraryServer01.

```
PS C:\> $FileShare = Get-SCStorageFileShare -Name "FileShare01"
```

```
PS C:\> $LibServer = Get-SCLibraryServer -Computersname "LibraryServer01.Contoso.com"
```

```
PS C:\> Unregister-SCStorageFileShare -StorageFileShare $FileShare -LibraryServer $LibServer
```

## **Related topics**

[Get-SCStorageFileShare](#)

[New-SCStorageFileShare](#)

---

[Register-SCStorageFileShare](#)

[Remove-SCStorageFileShare](#)

[Repair-SCStorageFileShare](#)

[Set-SCStorageFileShare](#)

---

# Unregister-SCStorageLogicalUnit

---

## Unregister-SCStorageLogicalUnit

Disassociates a logical unit from a host.

### Syntax

Parameter Set: RegisterToClusterJobGroup

```
Unregister-SCStorageLogicalUnit [-StorageLogicalUnit] <StorageLogicalUnit[]> -JobGroup <Guid> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

Parameter Set: RegisterToCluster

```
Unregister-SCStorageLogicalUnit [-StorageLogicalUnit] <StorageLogicalUnit[]> -VMHostCluster <HostCluster> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

Parameter Set: RegisterToHost

```
Unregister-SCStorageLogicalUnit [-StorageLogicalUnit] <StorageLogicalUnit[]> -VMHost <Host> [-JobGroup <Guid> ] [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

Parameter Set: RegisterToInitiators

```
Unregister-SCStorageLogicalUnit [-StorageLogicalUnit] <StorageLogicalUnit[]> -StorageInitiators <String[]> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

### Detailed Description

The Unregister-SCStorageLogicalUnit cmdlet disassociates a logical unit from a host.

### Parameters

#### **-JobGroup<Guid>**

Specifies an identifier for a series of commands that will run as a set just before the final command that includes the same job group identifier runs.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

---

|                             |       |
|-----------------------------|-------|
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|



|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-StorageInitiators<String[]>**

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-StorageLogicalUnit<StorageLogicalUnit[]>**

Specifies a storage logical unit object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-VMHost<Host>**

Specifies a virtual machine host object. VMM supports Hyper-V hosts, VMware ESX hosts, and Citrix XenServer hosts.

For more information about each type of host, type: "Get-Help Add-SCVMHost -detailed". See the examples for a specific cmdlet to determine how that cmdlet uses this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VMHostCluster<HostCluster>**

Specifies a VMM host cluster object.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | true  |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Examples**

### **1: Hide a logical unit from a host.**

The first command gets the host object named VMHost01 and stores the object in the \$VMHost variable.

The second command gets the storage logical unit object named LUN01 and stores the object in the \$LU variable.

The last command unregisters LUN01 from VMHost01.

```
PS C:\> $VMHost = Get-SCVMHost -ComputerName "VMHost01"
```

```
PS C:\> $LU = Get-SCStorageLogicalUnit -Name "LUN01"
```

```
PS C:\> Unregister-SCStorageLogicalUnit -StorageLogicalUnit $LU -VMHost $VMHost
```

---

## Related topics

[Get-SCStorageLogicalUnit](#)

[New-SCStorageLogicalUnit](#)

[Register-SCStorageLogicalUnit](#)

[Remove-SCStorageLogicalUnit](#)

[Set-SCStorageLogicalUnit](#)

---

# Update-SCService

---

## Update-SCService

Updates a VMM service instance.

### Syntax

Parameter Set: Service

```
Update-SCService [-Service] <Service> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-ShowActions] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

Parameter Set: WhatIf

```
Update-SCService [-Service] <Service> -WhatIf [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Update-SCService cmdlet updates an instance of a Virtual Machine Manager (VMM) service. There are two servicing types you can use: conventional servicing and image-based servicing.

Conventional servicing applies updates to deployed virtual machines in place, without redeploying the service. While fast, it does not allow changing a virtual hard disk, removing network adapters, or changing operating system settings (except for Windows Server roles and features).

Image-based servicing deploys new virtual machines to the service with the updates. This type of servicing is used most often after updating the VHD for a tier, such as applying software updates to the program disk.

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                        |       |
|------------------------|-------|
| Aliases                | none  |
| Required?              | false |
| Position?              | named |
| Default Value          | none  |
| Accept Pipeline Input? | false |

|                             |       |
|-----------------------------|-------|
| Accept Wildcard Characters? | false |
|-----------------------------|-------|

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-Service<Service>**

Specifies a VMM service object.

|                        |                |
|------------------------|----------------|
| Aliases                | none           |
| Required?              | true           |
| Position?              | 1              |
| Default Value          | none           |
| Accept Pipeline Input? | True (ByValue) |

|                             |       |
|-----------------------------|-------|
| Accept Wildcard Characters? | false |
|-----------------------------|-------|

## **-ShowActions**

Displays all servicing and orchestration actions that will be performed. This parameter is useful for debugging.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

|               |       |
|---------------|-------|
| Aliases       | none  |
| Required?     | true  |
| Position?     | named |
| Default Value | none  |

---

|                             |       |
|-----------------------------|-------|
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **Service**
- **ServicingActionItem[]**

## Notes

- Requires a VMM service object, which can be retrieved using the Get-SCService cmdlet.

## Examples

### 1: Update a service by using conventional servicing.

The first command gets the service object named Service01, which is a deployed service, and stores the object in the \$Service variable.

The second command gets the service template object named ServiceTemplate01 and stores the object in the \$SvcTemplate variable.

The third command creates an RTM release of the service template stored in \$SvcTemplate.

The fifth command sets the pending template on the service instance to the updated service template stored in \$PendingTemplate.

The last command updates Service01.

```
PS C:\> $Service = Get-SCService -Name "Service01"
```

```
PS C:\> $SvcTemplate = Get-SCServiceTemplate -Name "ServiceTemplate01"
```

```
PS C:\> $PendingTemplate = New-SCServiceTemplate -ServiceTemplate $SvcTemplate -Name  
"ServiceTemplate01" -Release "RTM"
```

```
PS C:\> Set-SCService -Service $Service -PendingServiceTemplate $PendingTemplate
```

```
PS C:\> Update-SCService -Service $Service
```

### 2: Update a service by using image-based servicing.

The first command gets the service object named Service02 and stores the object in the \$Service variable.

---

The second command gets the service template object named ServiceTemplate02 and stores the object in the \$SvcTemplate variable.

The third command creates a new release of the service template stored in \$SvcTemplate, names it ServiceTemplate02, gives it a release of RTM and stores the template in \$PendingTemplate.

The fourth command gets the computer tier object named Web Tier for the service template stored in \$PendingTemplate and stores the object in the \$WebTier variable.

The fifth command adds memory to the virtual machine template for the computer tier stored in \$WebTier.

The sixth command gets the virtual hard disk object named Win2k8R2BaseDisk\_Patched.vhd and stores the object in the \$BaseDisk2 variable. This virtual hard disk contains an updated version of the operating system.

The seventh command gets the virtual disk drive object on the virtual machine template stored in \$WebTemplate and stores the object in the \$VHD variable.

The eighth command removes the virtual disk drive object stored in \$VHD.

The ninth command adds the virtual hard disk object stored in \$BaseDisk2 to the virtual machine template object stored in \$WebTemplate.

The tenth command sets the pending template on the service instance to the updated service template stored in \$PendingTemplate.

The last command updates Service02.

```
PS C:\> $Service = Get-SCService -Name "Service02"
```

```
PS C:\> $SvcTemplate = Get-SCServiceTemplate -Name "ServiceTemplate02"
```

```
PS C:\> $PendingTemplate = New-SCServiceTemplate -ServiceTemplate $SvcTemplate -Name "ServiceTemplate02" -Release "RTM"
```

```
PS C:\> $WebTier = Get-SCComputerTierTemplate -ServiceTemplate $PendingTemplate -Name "Web Tier"
```

```
PS C:\> $WebTemplate = Get-SCVMTemplate -ComputerTierTemplate $WebTier | Set-SCVMTemplate -MemoryMB 2048
```

```
PS C:\> $BaseDisk2 = Get-SCVirtualHardDisk -Name "Win2k8R2BaseDisk_Patched.vhd"
```

```
PS C:\> $VHD = Get-SCVirtualDiskDrive -VMTemplate $WebTemplate
```

```
PS C:\> Remove-SCVirtualDiskDrive -VirtualDiskDrive $VHD
```

```
PS C:\> New-SCVirtualDiskDrive -VirtualHardDisk $BaseDisk2 -VMTemplate $WebTemplate -BootVolume -SystemVolume -Bus 0 -LUN 0 -ide -VolumeType BootAndSystem
```

```
PS C:\> Set-SCService -Service $Service -PendingServiceTemplate $PendingTemplate
```

```
PS C:\> Update-SCService -Service $Service
```

## Related topics

[Get-SCService](#)



---

# Update-SCServiceConfiguration

---

## Update-SCServiceConfiguration

Performs placement on a service configuration.

### Syntax

Parameter Set: Default

```
Update-SCServiceConfiguration -ServiceConfiguration <ServiceConfiguration> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Update-SCServiceConfiguration cmdlet performs placement on a service configuration object. During placement, the suitability of hosts is determined and configuration settings for the virtual machines are configured.

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-ServiceConfiguration<ServiceConfiguration>**

Specifies a service configuration object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

---

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **ServiceConfiguration**

## Examples

### 1: Run placement and update a service configuration.

The first command gets the service configuration object named Service01 and stores the object in the \$SvcConfig variable.

The second command updates and runs placement on service configuration Service01.

```
PS C:\> $SvcConfig = Get-SCServiceConfiguration -Name "Service01"
```

```
PS C:\> Update-SCServiceConfiguration -ServiceConfiguration $SvcConfig
```

## Related topics

[Get-SCServiceConfiguration](#)

[New-SCServiceConfiguration](#)

[Remove-SCServiceConfiguration](#)

[Set-SCServiceConfiguration](#)

---

# Update-SCVMConfiguration

---

## Update-SCVMConfiguration

Updates the properties of a VMM virtual machine configuration object.

### Syntax

Parameter Set: Default

```
Update-SCVMConfiguration [-VMConfiguration] <BaseVMConfiguration> [-JobVariable <String> ]  
[-PROTipID <Guid> ] [-RunAsynchronously] [-ValidateOnly <Boolean> ] [-VMName <String> ] [  
<CommonParameters>]
```

### Detailed Description

The Update-SCVMConfiguration cmdlet updates the properties of a Virtual Machine Manager (VMM) virtual machine configuration object.

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

---

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-ValidateOnly<Boolean>**

Indicates that validation of the placement will be performed, but placement will not actually be performed.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMConfiguration<BaseVMConfiguration>**

Specifies a virtual machine configuration object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **-VMName<String>**

Specifies the name of a virtual machine to be placed on a physical host server. Use this parameter to verify that another virtual machine with the same name is not already deployed on that host.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **ComputerConfig**

## **Notes**

- Requires a VMM virtual machine configuration object, which can be obtained by using the Get-SCVMConfiguration cmdlet.

---

## Examples

### 1: Update an existing virtual machine configuration.

The first command gets the service configuration object named Service01 from the VMM library and stores the object in the \$ServiceConfig variable.

The second command gets the virtual machine configuration object for the service configuration stored in \$ServiceConfig and stores the virtual machine configuration object in \$VMConfig.

The last command sets the ValidateOnly property to True for the first configuration object stored in \$VMConfig.

```
PS C:\> $ServiceConfig = Get-SCServiceConfiguration -Name "Service01"
```

```
PS C:\> $VMConfig = Get-SCVMConfiguration -ServiceConfiguration $ServiceConfig
```

```
PS C:\> Update-SCVMConfiguration -VMConfiguration $VMConfig[0] -ValidateOnly $True
```

### Related topics

[Get-SCVMConfiguration](#)

[New-SCVMConfiguration](#)

[Remove-SCVMConfiguration](#)

[Set-SCVMConfiguration](#)

---

# Update-SCVMMManagedComputer

---

## Update-SCVMMManagedComputer

Updates VMM agent software installed on a Windows-based managed computer.

### Syntax

Parameter Set: Default

```
Update-SCVMMManagedComputer [-VMMManagedComputer] <VMMManagedComputer> -Credential <VMMCredential> [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously] [ <CommonParameters>]
```

### Detailed Description

The Update-SCVMMManagedComputer cmdlet updates Virtual Machine Manager (VMM) agent software installed on a Windows-based managed computer to the current version of the software.

If you upgrade your VMM management server to a later version of the VMM service, afterward you can use this command to update agent software on computers that are managed by that VMM management server.

Managed computers that you can update by using this cmdlet include:

- Hyper-V hosts
- Windows-based library servers
- Windows-based P2V source computers

You can use the Update-SCVMMManagedComputer cmdlet to update the VMM agent software on domain-joined trusted hosts and non-trusted domain-joined hosts, but not on hosts located on a perimeter network.

### Parameters

#### **-Credential<VMMCredential>**

Specifies a credential object or, for some cmdlets, a Run As account object that contains the user name and password of an account that has permission to perform this action. Or, in the case of Restart-SCJob, has permission to complete a restarted task.

For more information about the PSCredential object, type: "Get-Help Get-Credential".

For more information about Run As accounts, type: "Get-Help New-SCRunAsAccount".



|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

## **-VMMManagedComputer<VMMManagedComputer>**

Specifies a computer object that is managed by VMM.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | true           |
| Position?                   | 1              |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Examples**

### **1: Update all managed computers.**

The first command prompts you to provide credentials with appropriate permissions to perform this operation and stores the credentials in the \$Credential variable.

The second command gets all computer objects that are currently managed by VMM and passes each object to "foreach" (the ForEach-Object cmdlet), which uses the Update-SCVMMManagedComputer cmdlet to update the agent software on each managed computer. As this command is processed, \$Credential provides your credentials to Update-SCVMMManagedComputer. Note: This example assumes that no managed computers are located in a perimeter network.

For more information about the standard Windows PowerShell ForEach-Object cmdlet, type: "Get-Help ForEach-Object".

```
PS C:\> $Credential = Get-Credential
```

---

```
PS C:\> Get-SCVMMManagedComputer | foreach { Update-SCVMMManagedComputer -VMMManagedComputer
$_ -Credential $Credential -RunAsynchronously }
```

## 2: Update a specific host.

The first command prompts you to provide credentials with appropriate permissions to perform this operation and stores the credentials in the `$Credential` variable.

The second command gets the managed host object named `VMHost01` and stores the object in the `$VMMManagedHost` variable.

The last command updates the agent software on `VMHost01`. As this command is processed, `$Credential` provides your credentials to `Update-SCVMMManagedComputer`.

```
PS C:\> $Credential = Get-Credential
```

```
PS C:\> $VMMManagedHost = Get-SCVMMManagedComputer -ComputerName "VMHost01.Contoso.com"
```

```
PS C:\> Update-SCVMMManagedComputer -VMMManagedComputer $VMMManagedHost -Credential
$Credential
```

## Related topics

[Get-SCVMMManagedComputer](#)

[Register-SCVMMManagedComputer](#)

---

# Write-SCOpsMgrConnection

---

## Write-SCOpsMgrConnection

Updates Operations Manager with the most current information from VMM.

### Syntax

Parameter Set: Default

```
Write-SCOpsMgrConnection [-JobVariable <String> ] [-PROTipID <Guid> ] [-RunAsynchronously]  
[-VMMServer <ServerConnection> ] [ <CommonParameters>]
```

### Detailed Description

The Write-SCOpsMgrConnection cmdlet updates Operations Manager with the most current information from Virtual Machine Manager (VMM).

### Parameters

#### **-JobVariable<String>**

Specifies that job progress is tracked and stored in the variable named by this parameter.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

#### **-PROTipID<Guid>**

Specifies the ID of the PRO tip that triggered this action. This allows for auditing of PRO tips.

|         |      |
|---------|------|
| Aliases | none |
|---------|------|

---

|                             |       |
|-----------------------------|-------|
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-RunAsynchronously**

Indicates that the job runs asynchronously so that control returns to the command shell immediately.

|                             |       |
|-----------------------------|-------|
| Aliases                     | none  |
| Required?                   | false |
| Position?                   | named |
| Default Value               | none  |
| Accept Pipeline Input?      | false |
| Accept Wildcard Characters? | false |

### **-VMMServer<ServerConnection>**

Specifies a VMM server object.

|                             |                |
|-----------------------------|----------------|
| Aliases                     | none           |
| Required?                   | false          |
| Position?                   | named          |
| Default Value               | none           |
| Accept Pipeline Input?      | True (ByValue) |
| Accept Wildcard Characters? | false          |

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

---

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **OpsMgrConnection**

## Examples

### 1: Update VMM data in Operations Manager.

This command updates Operations Manager with the most current data from VMM.

```
PS C:\> Write-SCOpsMgrConnection
```

## Related topics

[Get-SCOpsMgrConnection](#)

[New-SCOpsMgrConnection](#)

[Remove-SCOpsMgrConnection](#)

[Set-SCOpsMgrConnection](#)