

Microsoft System Center 2012 R2

Cmdlet Reference for System Center 2012 R2 Configuration Manager

Microsoft Corporation

Published: February 7, 2014

Applies To

System Center 2012 R2 Configuration Manager

Feedback

Send suggestions and comments about this document to sc2012docs@microsoft.com.

Copyright

This document is provided "as-is". Information and views expressed in this document, including URL and other Internet website 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.

© 2014 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
November 1, 2013	Initial release of this document.
February 7, 2014	Content improvements.

Contents

Add-CMApplicationCatalogWebServicePoint	24
Add-CMApplicationCatalogWebsitePoint	28
Add-CMAssetIntelligenceSynchronizationPoint	36
Add-CMBoundaryToGroup	40
Add-CMDeploymentType	45
Add-CMDeviceAffinityToUser	70
Add-CMDeviceCollectionDirectMembershipRule	74
Add-CMDeviceCollectionExcludeMembershipRule	78
Add-CMDeviceCollectionIncludeMembershipRule	83
Add-CMDeviceCollectionQueryMembershipRule	88
Add-CMDeviceCollectionToAdministrativeUser	92
Add-CMDeviceCollectionToDistributionPointGroup	97
Add-CMDistributionPoint	102
Add-CMDistributionPointToGroup	123
Add-CMDriverToDriverPackage	128
Add-CMEndpointProtectionPoint	133
Add-CMEnrollmentPoint	137
Add-CMEnrollmentProxyPoint	141
Add-CMFallbackStatusPoint	145
Add-CMManagementPoint	149
Add-CMOutOfBandServicePoint	155
Add-CMReportingServicePoint	161
Add-CMSecurityRoleToAdministrativeUser	166
Add-CMSecurityScopeToAdministrativeUser	171
Add-CMSoftwareUpdatePoint	176

Add-CMSoftwareUpdateToGroup	181
Add-CMStateMigrationPoint	186
Add-CMSystemHealthValidatorPoint.....	192
Add-CMUserAffinityToDevice	195
Add-CMUserCollectionDirectMembershipRule	199
Add-CMUserCollectionExcludeMembershipRule	203
Add-CMUserCollectionIncludeMembershipRule	208
Add-CMUserCollectionQueryMembershipRule	213
Add-CMUserCollectionToAdministrativeUser.....	217
Add-CMUserCollectionToDistributionPointGroup.....	222
Approve-CMApprovalRequest	227
Approve-CMDevice.....	231
Approve-CMUserDeviceAffinityRequest.....	234
Block-CMCertificate	239
Block-CMConflictingRecord.....	241
Block-CMDevice	245
Clear-CMAmtAuditLog.....	248
Clear-CMClientOperation	253
Clear-CMComponentStatusMessageCount	255
Clear-CMMigrationData	259
Clear-CMOperatingSystemImageUpdateSchedule.....	262
Clear-CMPxeDeployment	266
Clear-CMSiteStatusMessageCount.....	270
Convert-CMSchedule	273
Copy-CMClientAuthCertificateProfileConfigurationItem	275
Copy-CMRemoteConnectionProfileConfigurationItem	279
Copy-CMSecurityRole	283

Copy-CMTrustedRootCertificateProfileConfigurationItem	287
Copy-CMUserDataAndProfileConfigurationItem	291
Copy-CMVpnProfileConfigurationItem	295
Copy-CMWirelessProfileConfigurationItem	299
Deny-CMApprovalRequest	303
Deny-CMUserDeviceAffinityRequest.....	307
Disable-CMAlert.....	312
Disable-CMAmtAuditLog	315
Disable-CMBaseline	319
Disable-CMDriver	322
Disable-CMProgram	326
Disable-CMSoftwareMeteringRule	329
Disable-CMSoftwareUpdateAutoDeploymentRule	333
Disable-CMStatusFilterRule	337
Disable-CMTaskSequence	340
Enable-CMAlert	344
Enable-CMAmtAuditLog	347
Enable-CMAutomaticAMTProvisioning	351
Enable-CMBaseline	354
Enable-CMDriver	357
Enable-CMProgram	361
Enable-CMSoftwareMeteringRule	364
Enable-CMSoftwareUpdateAutoDeploymentRule.....	368
Enable-CMStatusFilterRule	372
Enable-CMTaskSequence.....	375
Export-CMAntimalwarePolicy	379
Export-CMApplication	383

Export-CMBaseline.....	388
Export-CMConfigurationItem.....	392
Export-CMDeviceCollection.....	396
Export-CMDriverPackage.....	400
Export-CMPackage.....	405
Export-CMSecurityRole.....	410
Export-CMTaskSequence.....	413
Export-CMUserCollection.....	418
Get-CMAccessAccount.....	422
Get-CMAccessLicense.....	433
Get-CMAccount.....	436
Get-CMActiveDirectoryForest.....	438
Get-CMActiveDirectorySite.....	441
Get-CMAministrativeUser.....	444
Get-CMAlert.....	447
Get-CMAlertSubscription.....	449
Get-CMAntiMalwarePolicy.....	452
Get-CMApplication.....	456
Get-CMApplicationCatalogWebServicePoint.....	458
Get-CMApplicationCatalogWebsitePoint.....	460
Get-CMApplicationRevisionHistory.....	462
Get-CMApprovalRequest.....	465
Get-CMAppV5XDeploymentTypeItem.....	468
Get-CMAppVVirtualEnvironment.....	470
Get-CMAssetIntelligenceCatalogItem.....	473
Get-CMAssetIntelligenceSynchronizationPoint.....	475
Get-CMAutomaticAmtProvisioningStatus.....	477

Get-CMBaseline	480
Get-CMBaselineSummarizationSchedule	484
Get-CMBaselineXMLDefinition	485
Get-CMBootImage	488
Get-CMBoundary	491
Get-CMBoundaryGroup	495
Get-CMCategory	498
Get-CMClientAuthCertificateProfileConfigurationItem	501
Get-CMClientOperations	504
Get-CMClientPushInstallation	505
Get-CMClientSetting	507
Get-CMClientStatusSetting	512
Get-CMClientStatusUpdateSchedule	514
Get-CMCloudDistributionPoint	516
Get-CMCollectionMembershipEvaluationComponent	520
Get-CMComponentStatusMessage	522
Get-CMComponentStatusSetting	526
Get-CMComputerAssociation	529
Get-CMConfigurationItem	532
Get-CMConfigurationItemHistory	535
Get-CMConfigurationItemXMLDefinition	538
Get-CMConflictingRecord	541
Get-CMDatabaseProperty	544
Get-CMDatabaseReplicationLinkProperty	546
Get-CMDataBaseReplicationStatus	548
Get-CMDeployment	550
Get-CMDeploymentPackage	552

Get-CMDeploymentStatus.....	554
Get-CMDeploymentType	556
Get-CMDevice	559
Get-CMDeviceCollection	564
Get-CMDeviceCollectionDirectMembershipRule.....	567
Get-CMDeviceCollectionExcludeMembershipRule	571
Get-CMDeviceCollectionIncludeMembershipRule	575
Get-CMDeviceCollectionQueryMembershipRule	579
Get-CMDeviceCollectionVariable	582
Get-CMDeviceVariable	585
Get-CMDiscoveryMethod	588
Get-CMDistributionPoint	591
Get-CMDistributionPointGroup	594
Get-CMDriver.....	596
Get-CMDriverPackage.....	600
Get-CMEmailNotificationComponent.....	603
Get-CMEndpointProtectionPoint	605
Get-CMEndpointProtectionSummarizationSchedule	607
Get-CMEnrollmentPoint.....	608
Get-CMEnrollmentProxyPoint.....	610
Get-CMExchangeServer.....	612
Get-CMFallbackStatusPoint	614
Get-CMFileReplicationRoute	616
Get-CMGlobalCondition.....	619
Get-CMHardwareRequirement	622
Get-CMInitModifiableSecuredCategory	624
Get-CMIPSubnet	626

Get-CMMaintenanceWindow.....	628
Get-CMManagementPoint.....	630
Get-CMManagementPointComponent.....	632
Get-CMMigrationCollection.....	634
Get-CMMigrationEntity.....	636
Get-CMMigrationEntityDependency.....	639
Get-CMOperatingSystemImage.....	641
Get-CMOperatingSystemImageUpdateSchedule.....	644
Get-CMOperatingSystemInstaller.....	647
Get-CMOutOfBandManagementComponent.....	650
Get-CMOutOfBandServicePoint.....	652
Get-CMPackage.....	654
Get-CMProgram.....	657
Get-CMQueryResultMaximum.....	659
Get-CMRemoteConnectionProfileConfigurationItem.....	661
Get-CMRemoteConnectionProfileConfigurationItemXmlDefinition.....	664
Get-CMReportingServicePoint.....	667
Get-CMSecurityRole.....	669
Get-CMSecurityScope.....	672
Get-CMSite.....	674
Get-CMSiteInstallStatus.....	676
Get-CMSiteMaintenanceTask.....	678
Get-CMSiteStatusMessage.....	680
Get-CMSiteSystemServer.....	683
Get-CMSoftwareDistributionComponent.....	685
Get-CMSoftwareInventory.....	687
Get-CMSoftwareMeteringRule.....	689

Get-CMSoftwareMeteringSetting.....	691
Get-CMSoftwareUpdate	693
Get-CMSoftwareUpdateAutoDeploymentRule	697
Get-CMSoftwareUpdateBasedClientInstallation.....	700
Get-CMSoftwareUpdateDeploymentPackage	702
Get-CMSoftwareUpdateGroup	705
Get-CMSoftwareUpdateLicense	708
Get-CMSoftwareUpdatePoint	711
Get-CMSoftwareUpdatePointComponent	713
Get-CMSoftwareUpdateSummarizationSchedule	715
Get-CMStateMigrationPoint.....	717
Get-CMStatusFilterRule.....	719
Get-CMStatusMessageQuery.....	721
Get-CMStatusReportingComponent.....	724
Get-CMStatusSummarizer.....	726
Get-CMSystemHealthValidatorPoint	728
Get-CMSystemHealthValidatorPointComponent.....	730
Get-CMTaskSequence	732
Get-CMTrustedRootCertificate	735
Get-CMTrustedRootCertificateProfileConfigurationItem	737
Get-CMUser.....	740
Get-CMUserCollection	743
Get-CMUserCollectionDirectMembershipRule	746
Get-CMUserCollectionExcludeMembershipRule.....	750
Get-CMUserCollectionIncludeMembershipRule.....	754
Get-CMUserCollectionQueryMembershipRule.....	758
Get-CMUserDataAndProfileConfigurationItem.....	761

Get-CMUserDataAndProfileConfigurationItemXmlDefinition	763
Get-CMUserDeviceAffinity	766
Get-CMUserDeviceAffinityRequest	769
Get-CMVhd	773
Get-CMVpnProfileConfigurationItem	776
Get-CMWindowsFirewallPolicy	779
Get-CMWirelessProfileConfigurationItem	781
Import-CMAntimalwarePolicy	784
Import-CMApplication	787
Import-CMBaseline	790
Import-CMCertificate	793
Import-CMComputerInformation	795
Import-CMConfigurationItem	801
Import-CMDeviceCollection	804
Import-CMDriver	806
Import-CMDriverPackage	812
Import-CMPackage	814
Import-CMSecurityRole	816
Import-CMSoftwareLicense	819
Import-CMTaskSequence	822
Import-CMUserCollection	825
Import-CMUserDeviceAffinity	827
Install-CMClient	830
Invoke-CMAmtProvisioningDiscovery	836
Invoke-CMBaselineSummarization	841
Invoke-CMClientNotification	843
Invoke-CMClientOperationSummarization	848

Invoke-CMContentValidation	850
Invoke-CMDeploymentSummarization	864
Invoke-CMDeviceCollectionUpdate	867
Invoke-CMDeviceRetire	870
Invoke-CMDeviceWipe	874
Invoke-CMEndpointProtectionScan	877
Invoke-CMEndpointProtectionSummarization	882
Invoke-CMForestDiscovery	884
Invoke-CMRemoteControl	886
Invoke-CMReport	891
Invoke-CMSecondarySiteUpgrade	895
Invoke-CMSoftwareUpdateAutoDeploymentRule	899
Invoke-CMSoftwareUpdateSummarization	902
Invoke-CMUserCollectionUpdate	904
Lock-CMObject	907
Merge-CMAntimalwarePolicy	910
Merge-CMConflictingRecord	914
Move-CMObject	918
New-CMAccessAccount	921
New-CMAccount	934
New-CMActiveDirectoryForest	937
New-CMAdministrativeUser	941
New-CMAlertSubscription	945
New-CMAmtProvisioningAccount	949
New-CMAntimalwarePolicy	952
New-CMApplication	956
New-CMAppVVirtualEnvironment	964

New-CMAssetIntelligenceCatalogItem	967
New-CMBaseline	971
New-CMBootImage	974
New-CMBoundary	978
New-CMBoundaryGroup	983
New-CMCategory	987
New-CMClientAuthCertificateProfileConfigurationItem	990
New-CMClientSetting	992
New-CMCloudDistributionPoint	996
New-CMComputerAssociation.....	1004
New-CMConfigurationItem	1008
New-CMDeviceCollection	1012
New-CMDeviceCollectionVariable.....	1017
New-CMDeviceVariable.....	1022
New-CMDistributionPointGroup.....	1026
New-CMDriverPackage	1029
New-CMExchangeClientAccessServer	1033
New-CMExchangeServer	1036
New-CMExchangeServerConnectorAccessRule	1044
New-CMExchangeServerConnectorApplicationSetting	1048
New-CMExchangeServerConnectorEmailManagementSetting	1051
New-CMExchangeServerConnectorGeneralSetting	1057
New-CMExchangeServerConnectorPasswordSetting	1061
New-CMExchangeServerConnectorSecuritySetting	1068
New-CMFileReplicationRoute.....	1074
New-CMGlobalCondition	1078
New-CMHardwareRequirement.....	1095

New-CMInstallationSourceFile	1099
New-CMMaintenanceWindow	1103
New-CMMigrationJob	1107
New-CMOperatingSystemImage	1116
New-CMOperatingSystemInstaller	1120
New-CMPackage	1123
New-CMProgram	1131
New-CMRemoteConnectionProfileConfigurationItem	1141
New-CMSchedule	1146
New-CMSecondarySite	1154
New-CMSecurityScope	1172
New-CMSiteSystemServer	1175
New-CMSoftwareMeteringRule	1181
New-CMSoftwareUpdateAutoDeploymentRule	1186
New-CMSoftwareUpdateGroup	1212
New-CMSqlServerSetting	1215
New-CMStatusFilterRule	1219
New-CMStatusMessageQuery	1229
New-CMStoragefolder	1232
New-CMTaskSequence	1236
New-CMTaskSequenceMedia	1255
New-CMTrustedRootCertificateProfileConfigurationItem	1273
New-CMUserCollection	1275
New-CMUserDataAndProfileConfigurationItem	1280
New-CMVhd	1309
New-CMVirtualEnvironmentGroup	1313
New-CMVpnProfileConfigurationItem	1316

New-CMWindowsFirewallPolicy	1318
New-CMWiredProfileObject.....	1326
New-CMWirelessProfileConfigurationItem	1330
New-CMWirelessProfileObject	1332
Publish-CMPrestageContent	1338
Publish-CMPrestageContentTaskSequence	1351
Remove-CMAccessAccount	1355
Remove-CMAccount.....	1368
Remove-CMActiveDirectoryForest	1371
Remove-CMAdministrativeUser	1375
Remove-CMAlert	1380
Remove-CMAlertSubscription	1384
Remove-CMAmtProvisioningData	1388
Remove-CMAntiMalwarePolicy	1393
Remove-CMApplication	1397
Remove-CMApplicationCatalogWebServicePoint.....	1401
Remove-CMApplicationCatalogWebSitePoint.....	1405
Remove-CMApplicationRevisionHistory	1409
Remove-CMAppVVirtualEnvironment	1413
Remove-CMAssetIntelligenceCatalogItem.....	1417
Remove-CMAssetIntelligenceSynchronizationPoint	1421
Remove-CMBaseline	1425
Remove-CMBootImage	1429
Remove-CMBoundary	1433
Remove-CMBoundaryFromGroup.....	1437
Remove-CMBoundaryGroup	1443
Remove-CMCategory	1447

Remove-CMClientAuthCertificateProfileConfigurationItem	1451
Remove-CMClientOperation.....	1455
Remove-CMClientSetting	1458
Remove-CMCloudDistributionPoint	1462
Remove-CMComputerAssociation	1466
Remove-CMConfigurationItem	1470
Remove-CMContentDistribution	1474
Remove-CMDeployment	1488
Remove-CMDeploymentType	1492
Remove-CMDevice.....	1496
Remove-CMDeviceAffinityFromUser.....	1500
Remove-CMDeviceCollection.....	1504
Remove-CMDeviceCollectionDirectMembershipRule	1508
Remove-CMDeviceCollectionExcludeMembershipRule	1513
Remove-CMDeviceCollectionFromAdministrativeUser	1518
Remove-CMDeviceCollectionFromDistributionPointGroup.....	1523
Remove-CMDeviceCollectionIncludeMembershipRule.....	1527
Remove-CMDeviceCollectionQueryMembershipRule	1532
Remove-CMDeviceCollectionVariable	1536
Remove-CMDeviceVariable	1540
Remove-CMDistributionPoint	1544
Remove-CMDistributionPointFromGroup.....	1548
Remove-CMDistributionPointGroup	1553
Remove-CMDriver	1557
Remove-CMDriverFromDriverPackage.....	1561
Remove-CMDriverPackage.....	1566
Remove-CMEndpointProtectionPoint.....	1570

Remove-CMEnrollmentPoint	1574
Remove-CMEnrollmentProxyPoint	1578
Remove-CMExchangeServer	1582
Remove-CMFallbackStatusPoint	1585
Remove-CMFileReplicationRoute	1589
Remove-CMGlobalCondition	1592
Remove-CMHardwareRequirement	1596
Remove-CMMaintenanceWindow	1599
Remove-CMManagementPoint	1602
Remove-CMOperatingSystemImage	1606
Remove-CMOperatingSystemInstaller	1610
Remove-CMOutOfBandServicePoint	1614
Remove-CMPackage	1618
Remove-CMProgram	1622
Remove-CMRemoteConnectionProfileConfigurationItem	1626
Remove-CMReportingServicePoint	1629
Remove-CMSecondarySite	1633
Remove-CMSecurityRole	1637
Remove-CMSecurityRoleFromAdministrativeUser	1641
Remove-CMSecurityScope	1646
Remove-CMSecurityScopeFromAdministrativeUser	1650
Remove-CMSoftwareMeteringRule	1655
Remove-CMSoftwareUpdateAutoDeploymentRule	1659
Remove-CMSoftwareUpdateDeploymentPackage	1663
Remove-CMSoftwareUpdateGroup	1667
Remove-CMSoftwareUpdatePoint	1671
Remove-CMStateMigrationPoint	1675

Remove-CMStatusFilterRule	1679
Remove-CMStatusMessageQuery	1683
Remove-CMSystemHealthValidatorPoint.....	1687
Remove-CMTaskSequence.....	1691
Remove-CMTrustedRootCertificateProfileConfigurationItem.....	1695
Remove-CMUser	1699
Remove-CMUserAffinityFromDevice.....	1703
Remove-CMUserCollection	1707
Remove-CMUserCollectionDirectMembershipRule	1711
Remove-CMUserCollectionExcludeMembershipRule	1716
Remove-CMUserCollectionFromAdministrativeUser	1721
Remove-CMUserCollectionFromDistributionPointGroup	1726
Remove-CMUserCollectionIncludeMembershipRule	1730
Remove-CMUserCollectionQueryMembershipRule	1735
Remove-CMUserDataAndProfileConfigurationItem	1739
Remove-CMVhd	1743
Remove-CMVpnProfileConfigurationItem	1747
Remove-CMWindowsFirewallPolicy.....	1750
Remove-CMWirelessProfileConfigurationItem	1754
Resolve-CMInventoriedSoftwareConflict.....	1757
Restore-CMApplicationRevisionHistory.....	1760
Resume-CMApplication	1764
Save-CMDatabaseReplicationDiagnostic.....	1767
Save-CMEndpointProtectionDefinition	1770
Save-CMSoftwareUpdate	1775
Send-CMAssetIntelligenceCatalogUpdateRequest.....	1781
Set-CMAccessAccount	1784

Set-CMAccount.....	1797
Set-CMActiveDirectoryForest	1801
Set-CMAlert	1805
Set-CMAlertSubscription	1810
Set-CMAntiMalwarePolicy	1817
Set-CMApplication	1842
Set-CMApplicationCatalogWebsitePoint	1855
Set-CMApplicationDeployment.....	1863
Set-CMAppVVirtualEnvironment	1876
Set-CMAssetIntelligenceCatalogItem	1882
Set-CMAssetIntelligenceClass	1887
Set-CMAssetIntelligenceSynchronizationPoint	1892
Set-CMAssignedSite.....	1895
Set-CMBaseline.....	1899
Set-CMBaselineDeployment.....	1907
Set-CMBaselineSummarizationSchedule.....	1914
Set-CMBootImage	1917
Set-CMBoundary	1924
Set-CMBoundaryGroup	1929
Set-CMClientAuthCertificateProfileConfigurationItem	1935
Set-CMClientPushInstallation	1939
Set-CMClientSetting	1945
Set-CMClientStatusSetting	1985
Set-CMClientStatusUpdateSchedule	1989
Set-CMCloudDistributionPoint.....	1992
Set-CMCollectionMembershipEvaluationComponent	1998
Set-CMComputerAssociation	2001

Set-CMConfigurationItem	2006
Set-CMConfigurationPolicyDeployment	2013
Set-CMDatabaseProperty.....	2021
Set-CMDatabaseReplicationLinkProperty	2025
Set-CMDeploymentType	2034
Set-CMDeviceCollection.....	2055
Set-CMDeviceCollectionVariable	2060
Set-CMDeviceOwnership	2065
Set-CMDeviceVariable	2069
Set-CMDiscoveryMethod.....	2073
Set-CMDistributionPoint	2085
Set-CMDistributionPointGroup	2105
Set-CMDriver	2110
Set-CMDriverBootImage.....	2119
Set-CMDriverPackage	2125
Set-CMEmailNotificationComponent	2131
Set-CMEndpointProtectionPoint.....	2136
Set-CMEndpointProtectionSummarizationSchedule	2140
Set-CMEnrollmentPoint	2143
Set-CMExchangeServer	2147
Set-CMFallbackStatusPoint.....	2158
Set-CMFileReplicationRoute	2163
Set-CMGlobalCondition	2172
Set-CMHardwareRequirement	2189
Set-CMHierarchySetting	2194
Set-CMMaintenanceWindow	2199
Set-CMManagementPoint	2203

Set-CMManagementPointComponent.....	2211
Set-CMMigrationExclusionList.....	2215
Set-CMMigrationSource	2217
Set-CMOperatingSystemImage.....	2221
Set-CMOperatingSystemImageUpdateSchedule	2228
Set-CMOperatingSystemInstaller	2234
Set-CMOutOfBandManagementComponent.....	2241
Set-CMOutOfBandServicePoint	2255
Set-CMPackage.....	2261
Set-CMPackageDeployment	2272
Set-CMPowerControl	2289
Set-CMProgram.....	2293
Set-CMQueryResultMaximum	2308
Set-CMRemoteConnectionProfileConfigurationItem	2310
Set-CMReportingServicePoint.....	2316
Set-CMSecurityRole	2321
Set-CMSecurityScope	2325
Set-CMSite.....	2329
Set-CMSiteMaintenanceTask	2349
Set-CMSiteSystemServer.....	2359
Set-CMSoftwareDistributionComponent.....	2365
Set-CMSoftwareInventory.....	2370
Set-CMSoftwareMeteringRule	2375
Set-CMSoftwareMeteringSetting	2382
Set-CMSoftwareUpdate.....	2386
Set-CMSoftwareUpdateAutoDeploymentRule.....	2390
Set-CMSoftwareUpdateBasedClientInstallation	2417

Set-CMSoftwareUpdateDeployment	2420
Set-CMSoftwareUpdateDeploymentPackage	2438
Set-CMSoftwareUpdateGroup	2445
Set-CMSoftwareUpdatePoint	2451
Set-CMSoftwareUpdatePointComponent	2458
Set-CMSoftwareUpdateSummarizationSchedule	2468
Set-CMStateMigrationPoint	2471
Set-CMStatusFilterRule	2479
Set-CMStatusMessageQuery	2489
Set-CMStatusReportingComponent	2495
Set-CMStatusSummarizer	2504
Set-CMSystemHealthValidatorPointComponent	2512
Set-CMTaskSequence	2519
Set-CMTaskSequenceDeployment	2530
Set-CMTrustedRootCertificateProfileConfigurationItem	2546
Set-CMUserCollection	2550
Set-CMUserDataAndProfileConfigurationItem	2555
Set-CMVhd	2586
Set-CMVpnProfileConfigurationItem	2593
Set-CMWindowsFirewallPolicy	2596
Set-CMWirelessProfileConfigurationItem	2607
Start-CMAntimalwarePolicyDeployment	2610
Start-CMApplicationDeployment	2616
Start-CMApplicationDeploymentSimulation	2629
Start-CMBaselineDeployment	2634
Start-CMClientSettingDeployment	2641
Start-CMCloudDistributionPoint	2646

Start-CMConfigurationPolicyDeployment	2650
Start-CMContentDistribution	2659
Start-CMDistributionPointUpgrade	2675
Start-CMPackageDeployment	2694
Start-CMSoftwareUpdateDeployment	2711
Start-CMTaskSequenceDeployment	2729
Stop-CMCloudDistributionPoint	2744
Suspend-CMAlert	2748
Suspend-CMApplication	2752
Sync-CMAssetIntelligenceCatalog	2755
Sync-CMExchangeServer	2757
Sync-CMSoftwareUpdate	2760
Unblock-CMCertificate	2762
Unblock-CMDevice	2764
Unblock-CMThreat.....	2767
Undo-CMSoftwareInventory	2772
Unlock-CMObject.....	2775
Update-CMAMTProvisioning	2778
Update-CMApplicationStatistic	2782
Update-CMCertificate	2785
Update-CMClientStatus	2788
Update-CMDistributionPoint	2790

Add-CMApplicationCatalogWebServicePoint

Add-CMApplicationCatalogWebServicePoint

Adds an Application Catalog web service point to Configuration Manager.

Syntax

Parameter Set: AppWebServiceP

```
Add-CMApplicationCatalogWebServicePoint -SiteCode <String> -SiteSystemServerName <String> [-CommunicationType {HTTP | HTTPS} ] [-IISWebsite <String> ] [-PortNumber <Int32> ] [-WebApplicationName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Add-CMApplicationCatalogWebServicePoint** cmdlet adds an Application Catalog web service point to a Microsoft System Center 2012 Configuration Manager site.

System Center 2012 Configuration Manager requires a web service point site system role to support the Application Catalog website and the Software Library. You also need an Application Catalog website point in the same site, but not necessarily on the same server. If you intend to use Secure Hypertext Transfer Protocol (HTTPS), you need to deploy a web server certificate on the server hosting the web service point. For more information about site system roles, see [Install and Configure Site System Roles for Configuration Manager](http://go.microsoft.com/fwlink/?LinkId=262649) (http://go.microsoft.com/fwlink/?LinkId=262649) on TechNet.

Parameters

-CommunicationType<ComputerCommunicationType>

Specifies the communication type. Valid values are: HTTP and HTTPS.

The acceptable values for this parameter are:

HTTP	
HTTPS	

Aliases	none
Required?	false
Position?	named
Default Value	none

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

-IISWebsite<String>

Specifies the Internet Information Services (IIS) website installed on the Application Catalog web service point server.

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

-PortNumber<Int32>

Specifies the port to use to connect with the web service.

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

-SiteCode<String>

Specifies the site code for a Configuration Manager site.

Aliases	none
Required?	true
Position?	named

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

-SiteSystemServerName<String>

Specifies the name of a server that hosts a site system role.

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

-WebApplicationName<String>

Specifies the name of the web application used for the application catalog.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Add a web service point for the Application Catalog

This command adds a web service point for the Application Catalog. The command specifies a port for the service. The command specifies the site code for the site that the role belongs to, as well as the name of the server that hosts the role.

```
PS C:\> Add-CMApplicationCatalogWebServicePoint -PortNumber 80 -SiteCode "CM1" -
SiteSystemServerName "CMACWSPRole.Western.Contoso.com"
```

Related topics

[Get-CMApplicationCatalogWebServicePoint](#)

[Remove-CMApplicationCatalogWebServicePoint](#)

[Add-CMApplicationCatalogWebsitePoint](#)

[Get-CMApplicationCatalogWebsitePoint](#)

Add-CMApplicationCatalogWebsitePoint

Add-CMApplicationCatalogWebsitePoint

Adds an Application Catalog website point to a Configuration Manager site.

Syntax

Parameter Set: AppWebSiteP

```
Add-CMApplicationCatalogWebsitePoint -ConfiguredAsHttpConnection -SiteCode <String> -  
SiteSystemServerName <String> -  
SiteSystemServerNameConfiguredForApplicationCatalogWebServicePoint <String> [-ColorBlue  
<Int32> ] [-ColorGreen <Int32> ] [-ColorRed <Int32> ] [-IISWebsite <String> ] [-NetbiosName  
<String> ] [-OrganizationName <String> ] [-PortForHttpConnection <Int32> ] [-  
PortForHttpsConnection <Int32> ] [-WebApplicationName <String> ] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: AppWebSitePWithSSL

```
Add-CMApplicationCatalogWebsitePoint -ConfiguredAsHttpsConnection -SiteCode <String> -  
SiteSystemServerName <String> -  
SiteSystemServerNameConfiguredForApplicationCatalogWebServicePoint <String> [-  
ClientConnectionType {Internet | InternetAndIntranet | Intranet} ] [-ColorBlue <Int32> ] [-  
ColorGreen <Int32> ] [-ColorRed <Int32> ] [-IISWebsite <String> ] [-NetbiosName <String> ]  
[-OrganizationName <String> ] [-PortForHttpConnection <Int32> ] [-PortForHttpsConnection  
<Int32> ] [-WebApplicationName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Add-CMApplicationCatalogWebsitePoint** cmdlet adds an Application Catalog website point to a Microsoft System Center 2012 Configuration Manager site. This site system role supports the Application Catalog website and the Software Library.

Specify the site that this website point supports and the server that hosts the website point. You can specify the website name and NetBIOS name of the Application Catalog. You can also specify port numbers for HTTP and HTTPS.

You can customize the page that users see when they connect to the Application Catalog. Specify custom values for the colors blue, green, and red. You can also specify a name for users to see in the browser, such as a company name or a division within a company.

Parameters

-ClientConnectionType<ClientConnectionTypes>

Specifies how a client connects to the website. Valid values are:

- Internet
- InternetAndIntranet
- Intranet

The acceptable values for this parameter are:

Internet	
InternetAndIntranet	
Intranet	

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

-ColorBlue<Int32>

Specifies an integer value for a custom blue color. Configuration Manager uses custom colors to conform to customer branding.

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

-ColorGreen<Int32>

Specifies an integer value for a custom green color. Configuration Manager uses custom colors to conform to customer branding.

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

-ColorRed<Int32>

Specifies an integer value for a custom red color. Configuration Manager uses custom colors to conform to customer branding.

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

-ConfiguredAsHttpConnection

Indicates that clients connect to the website by using HTTP. To require clients to connect by using HTTPS, specify the *ConfiguredAsHttpsConnection* parameter instead of this parameter.

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

-ConfiguredAsHttpsConnection

Indicates that client connect to the website by using HTTPS. To require clients to connect by using HTTP, specify the *ConfiguredAsHttpConnection* parameter instead of this parameter.

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

-IISWebsite<String>

Specifies the Internet Information Services (IIS) website installed on the Application Catalog website point server.

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

-NetbiosName<String>

Specifies the NetBIOS name of the server that hosts the Application Catalog website point.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

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

-OrganizationName<String>

Specifies a name for a customer organization. This name appears to users who access the Application Catalog.

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

-PortForHttpConnection<Int32>

Specifies a port for HTTP. Users connect to the website point by using this port. If you do not specify a value for this parameter, the cmdlet uses the default port of 80.

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

-PortForHttpsConnection<Int32>

Specifies a port for HTTPS. Users connect to the website point by using this port. If you do not specify a value for this parameter, the cmdlet uses the default port of 443.

Aliases	none
Required?	false
Position?	named

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

-SiteCode<String>

Specifies the site code for a Configuration Manager site.

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

-SiteSystemServerName<String>

Specifies the name of a server that hosts a site system role.

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

-

SiteSystemServerNameConfiguredForApplicationCatalogWebServicePoint<String>

Specifies the name of a server that hosts the site system role for the Application Catalog web service point.

The Application Catalog web service point is a site system role that provides information about available software from the Software Library to the Application Catalog website.

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

-WebApplicationName<String>

Specifies the name of the web application used for the application catalog.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Add an Application Catalog website point

This command adds an Application Catalog website point site system role for the site that has the site code CM4. A server named ApplicationCatalog.Western.Contoso.com hosts the website point. The command specifies values for the three colors.

```
PS C:\> Add-CMApplicationCatalogWebsitePoint -ColorBlue 52 -ColorGreen 201 -ColorRed 168 -  
SiteCode "CM4" -SiteSystemServerName "ApplicationCatalog.Western.Contoso.com"
```

Related topics

[Get-CMApplicationCatalogWebsitePoint](#)

[Remove-CMApplicationCatalogWebSitePoint](#)

[Set-CMApplicationCatalogWebsitePoint](#)

Add-CMAssetIntelligenceSynchronizationPoint

Add-CMAssetIntelligenceSynchronizationPoint

Installs an Asset Intelligence synchronization point.

Syntax

Parameter Set: AISyncPoint

```
Add-CMAssetIntelligenceSynchronizationPoint -SiteSystemServerName <String> [-CertificateFile <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: AISyncPointWithSchedule

```
Add-CMAssetIntelligenceSynchronizationPoint -EnableSynchronization -ScheduleToken <IResultObject> -SiteSystemServerName <String> [-CertificateFile <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Add-CMAssetIntelligenceSynchronizationPoint** cmdlet installs an Asset Intelligence synchronization point. Microsoft System Center 2012 Configuration Manager uses the Asset Intelligence synchronization point site system role to connect System Center 2012 Configuration Manager sites to System Center Online to synchronize Asset Intelligence catalog information.

You can install the Asset Intelligence synchronization point only on a site system located at the top-level site of the System Center 2012 Configuration Manager hierarchy.

Parameters

-CertificateFile<String>

Specifies the path to a System Center Online authentication certificate (.pfx) file.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

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

-EnableSynchronization

Indicates that the Asset Intelligence synchronization point is enabled.

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

-ScheduleToken<IResultObject>

Specifies a Configuration Manager schedule object. You can use the [New-CMSchedule](#) cmdlet to create a schedule token.

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

-SiteSystemServerName<String>

Specifies an array of fully qualified domain names (FQDN) of the servers that host the site system role.

Aliases	none
Required?	true
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Install an Asset Intelligence synchronization point

This command installs an Asset Intelligence synchronization point on the site system server named CMDIV-TSQA04.CORP.CONTOSO.COM.

```
PS C:\> Add-CMAssetIntelligenceSynchronizationPoint -SiteSystemServerName "CMDIV-TSQA04.CORP.CONTOSO.COM"
```

Example 2: Install a scheduled Asset Intelligence synchronization point

This first command creates a System Center 2012 Configuration Manager schedule token that specifies an event that occurs once a week for three weeks on Fridays. The command stores the results in the **\$Sc** variable.

The second command installs an Asset Intelligence synchronization point on the site system server named CMDIV-TSQA04.CORP.CONTOSO.COM, specifying the schedule stored in \$Sc. The command also specifies the System Center Online authentication certificate (.pfx) file, and enables synchronization.

```
PS C:\> $Sc = New-CMSchedule -DayOfWeek Friday -RecurCount 2
PS C:\> Add-CMAssetIntelligenceSynchronizationPoint -SiteSystemServerName "CMDIV-
TSQA04.CORP.CONTOSO.COM" -CertificateFile "\\Contoso01\CM\ACDataFile\AIpfx.pfx" -
EnableSynchronization -ScheduleToken $Sc
```

Related topics

[Get-CMAssetIntelligenceSynchronizationPoint](#)

[Remove-CMAssetIntelligenceSynchronizationPoint](#)

[Set-CMAssetIntelligenceSynchronizationPoint](#)

[New-CMSchedule](#)

Add-CMBoundaryToGroup

Add-CMBoundaryToGroup

Assigns boundaries to a boundary group.

Syntax

Parameter Set: AddBoundaryToGroupById_Id

```
Add-CMBoundaryToGroup -BoundaryGroupId <Int32> -BoundaryId <Int32> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: AddBoundaryToGroupById_Name

```
Add-CMBoundaryToGroup -BoundaryGroupName <String> -BoundaryId <Int32> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: AddBoundaryToGroupById_Object

```
Add-CMBoundaryToGroup -BoundaryGroup <IResultObject> -BoundaryId <Int32> [-Confirm] [-  
WhatIf] [ <CommonParameters>]
```

Parameter Set: AddBoundaryToGroupByName_Id

```
Add-CMBoundaryToGroup -BoundaryGroupId <Int32> -BoundaryName <String> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: AddBoundaryToGroupByName_Name

```
Add-CMBoundaryToGroup -BoundaryGroupName <String> -BoundaryName <String> [-Confirm] [-  
WhatIf] [ <CommonParameters>]
```

Parameter Set: AddBoundaryToGroupByName_Object

```
Add-CMBoundaryToGroup -BoundaryGroup <IResultObject> -BoundaryName <String> [-Confirm] [-  
WhatIf] [ <CommonParameters>]
```

Parameter Set: AddBoundaryToGroupByObject_Id

```
Add-CMBoundaryToGroup -Boundary <IResultObject> -BoundaryGroupId <Int32> [-Confirm] [-  
WhatIf] [ <CommonParameters>]
```

Parameter Set: AddBoundaryToGroupByObject_Name

```
Add-CMBoundaryToGroup -Boundary <IResultObject> -BoundaryGroupName <String> [-Confirm] [-  
WhatIf] [ <CommonParameters>]
```

Parameter Set: AddBoundaryToGroupByObject_Object

```
Add-CMBoundaryToGroup -Boundary <IResultObject> -BoundaryGroup <IResultObject> [-Confirm] [-  
WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Add-CMBoundaryToGroup** cmdlet assigns boundaries to a boundary group.

In Microsoft System Center 2012 Configuration Manager, a boundary is an intranet location that contains one or more devices that you can manage. A boundary can be an IP subnet, Active Directory site name, IPv6 prefix, or an IP address range.

You can use boundary groups to manage network locations. You must assign boundaries to boundary groups before you can use the boundary group. Boundary groups enable client computers to find a primary site for client assignment, which is referred to as automatic site assignment, and a list of available site systems that have content. For more information about boundaries, see [Planning for Boundaries and Boundary Groups in Configuration Manager](http://go.microsoft.com/fwlink/?LinkId=266225) (<http://go.microsoft.com/fwlink/?LinkId=266225>) on TechNet.

Parameters

-Boundary<IResultObject>

Specifies a boundary object. To obtain a **CMBoundary** object, use the **Get-CMBoundary** cmdlet.

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

-BoundaryGroup<IResultObject>

Specifies a boundary group object. To obtain a **CMBoundaryGroup** object, use the **Get-CMBoundaryGroup** cmdlet.

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

-BoundaryGroupId<Int32>

Specifies the ID of a boundary group.

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

-BoundaryGroupName<String>

Specifies the name of a boundary group.

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

-BoundaryId<Int32>

Specifies the ID of a boundary.

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

-BoundaryName<String>

Specifies the name of a boundary.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Assign a boundary group to a boundary

This command assigns the boundary named to CLBound03 to the boundary group that has the Id 16777219.

```
PS C:\> Add-CMBoundaryToGroup -BoundaryGroupID "16777219" -BoundaryName "CLBound03"
```

Related topics

[Get-CMBoundary](#)

[Get-CMBoundaryGroup](#)

[Remove-CMBoundaryFromGroup](#)

Add-CMDeploymentType

Add-CMDeploymentType

Adds a deployment type for an application.

Syntax

Parameter Set: AddDeploymentTypeByAndroidDeepLinkInstallerAuto

```
Add-CMDeploymentType -AndroidDeepLinkInstaller -ApplicationName <String> -
AutoIdentifyFromInstallationFile -ForceForUnknownPublisher <Boolean> -
InstallationFileLocation <String> [-AdministratorComment <String> ] [-DeploymentTypeName
<String> ] [-Language <String[]> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: AddDeploymentTypeByAndroidInstallerAuto

```
Add-CMDeploymentType -AndroidInstaller -ApplicationName <String> -
AutoIdentifyFromInstallationFile -ForceForUnknownPublisher <Boolean> -
InstallationFileLocation <String> [-AdministratorComment <String> ] [-DeploymentTypeName
<String> ] [-Language <String[]> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: AddDeploymentTypeByAppV5xInstallerAuto

```
Add-CMDeploymentType -ApplicationName <String> -AppV5xInstaller -
AutoIdentifyFromInstallationFile -ForceForUnknownPublisher <Boolean> -
InstallationFileLocation <String> [-AdministratorComment <String> ] [-
AllowClientsToUseFallbackSourceLocationForContent <Boolean> ] [-DeploymentTypeName <String>
] [-Language <String[]> ] [-OnSlowNetworkMode {DoNothing | Download |
DownloadContentForStreaming} ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: AddDeploymentTypeByAppvInstallerAuto

```
Add-CMDeploymentType -ApplicationName <String> -AppvInstaller -
AutoIdentifyFromInstallationFile -ForceForUnknownPublisher <Boolean> -
InstallationFileLocation <String> [-AdministratorComment <String> ] [-
AllowClientsToUseFallbackSourceLocationForContent <Boolean> ] [-DeploymentTypeName <String>
] [-Language <String[]> ] [-OnFastNetworkMode {RunFromNetwork | RunLocal} ] [-
OnSlowNetworkMode {DoNothing | Download | DownloadContentForStreaming} ] [-Confirm] [-
WhatIf] [ <CommonParameters>]
```

Parameter Set: AddDeploymentTypeByDeeplinkInstallerAuto

```
Add-CMDeploymentType -ApplicationName <String> -AutoIdentifyFromInstallationFile -
DeeplinkInstaller -ForceForUnknownPublisher <Boolean> [-AdministratorComment <String> ] [-
ApplicationNameInWindowsStore <String> ] [-DeploymentTypeName <String> ] [-
InstallationFileLocation <String> ] [-Language <String[]> ] [-RemoteComputerName <String> ]
[-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: AddDeploymentTypeByiOSDeepLinkInstallerAuto

```
Add-CMDeploymentType -ApplicationName <String> -AutoIdentifyFromInstallationFile -
ForceForUnknownPublisher <Boolean> -InstallationFileLocation <String> -iOSDeepLinkInstaller
[-AdministratorComment <String> ] [-DeploymentTypeName <String> ] [-Language <String[]> ] [-
```

Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: AddDeploymentTypeByIOSInstallerAuto

Add-CMDeploymentType -ApplicationName <String> -AutoIdentifyFromInstallationFile - ForceForUnknownPublisher <Boolean> -InstallationFileLocation <String> -IOSInstaller [- AdministratorComment <String>] [-DeploymentTypeName <String>] [-Language <String[]>] [- Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: AddDeploymentTypeByMacInstallerAuto

Add-CMDeploymentType -ApplicationName <String> -AutoIdentifyFromInstallationFile - ForceForUnknownPublisher <Boolean> -InstallationFileLocation <String> -MacInstaller [- AdministratorComment <String>] [-DeploymentTypeName <String>] [-Language <String[]>] [- Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: AddDeploymentTypeByMsiInstallerAuto

Add-CMDeploymentType -ApplicationName <String> -AutoIdentifyFromInstallationFile - ForceForUnknownPublisher <Boolean> -InstallationFileLocation <String> -MsiInstaller [- AdministratorComment <String>] [-AllowClientsToUseFallbackSourceLocationForContent <Boolean>] [-DeploymentTypeName <String>] [-InstallationBehaviorType {InstallForSystem | InstallForSystemIfResourceIsDeviceOtherwiseInstallForUser | InstallForUser}] [- InstallationProgram <String>] [-Language <String[]>] [-OnSlowNetworkMode {DoNothing | Download | DownloadContentForStreaming}] [- RunInstallationProgramAs32BitProcessOn64BitClient <Boolean>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: AddDeploymentTypeByMsiInstallerManual

Add-CMDeploymentType -ApplicationName <String> -DeploymentTypeName <String> - DetectDeploymentTypeByCustomScript -InstallationProgram <String> - ManualSpecifyDeploymentType -MsiInstaller -ScriptContent <String> -ScriptType <ScriptLanguage> [-AdministratorComment <String>] [-AllowClientsToShareContentOnSameSubnet <Boolean>] [-AllowClientsToUseFallbackSourceLocationForContent <Boolean>] [- ContentLocation <String>] [-EstimatedInstallationTimeMinutes <Int32>] [- InstallationBehaviorType {InstallForSystem | InstallForSystemIfResourceIsDeviceOtherwiseInstallForUser | InstallForUser}] [- InstallationProgramVisibility {Normal | Minimized | Maximized | Hidden}] [- InstallationStartIn <String>] [-Language <String[]>] [-LogonRequirementType {OnlyWhenNoUserLoggedIn | OnlyWhenUserLoggedIn | WhereOrNotUserLoggedIn | WhetherOrNotUserLoggedIn}] [-MaximumAllowedRunTimeMinutes <Int32>] [-OnSlowNetworkMode {DoNothing | Download | DownloadContentForStreaming}] [-PersistContentInClientCache <Boolean>] [-RequiresUserInteraction <Boolean>] [- RunInstallationProgramAs32BitProcessOn64BitClient <Boolean>] [- RunScriptAs32bitProcessOn64bitClient <Boolean>] [-UninstallProgram <String>] [- UninstallStartIn <String>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: AddDeploymentTypeByNokiaInstallerAuto

Add-CMDeploymentType -ApplicationName <String> -AutoIdentifyFromInstallationFile - ForceForUnknownPublisher <Boolean> -InstallationFileLocation <String> -NokiaInstaller [- AdministratorComment <String>] [-DeploymentTypeName <String>] [-Language <String[]>] [- Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: AddDeploymentTypeByNokiaInstallerManual

Add-CMDeploymentType -ApplicationName <String> -ContentLocation <String> -DeploymentTypeName

<String> -ManualSpecifyDeploymentType -NokiaInstaller [-AdministratorComment <String>] [-Language <String[]>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: AddDeploymentTypeByScriptInstallerManual

Add-CMDeploymentType -ApplicationName <String> -DeploymentTypeName <String> -DetectDeploymentTypeByCustomScript -InstallationProgram <String> -ManualSpecifyDeploymentType -ScriptContent <String> -ScriptInstaller -ScriptType <ScriptLanguage> [-AdministratorComment <String>] [-AllowClientsToShareContentOnSameSubnet <Boolean>] [-ContentLocation <String>] [-EstimatedInstallationTimeMinutes <Int32>] [-InstallationBehaviorType {InstallForSystem | InstallForSystemIfResourceIsDeviceOtherwiseInstallForUser | InstallForUser}] [-InstallationProgramVisibility {Normal | Minimized | Maximized | Hidden}] [-InstallationStartIn <String>] [-Language <String[]>] [-LogonRequirementType {OnlyWhenNoUserLoggedIn | OnlyWhenUserLoggedIn | WhereOrNotUserLoggedIn | WhetherOrNotUserLoggedIn}] [-MaximumAllowedRunTimeMinutes <Int32>] [-PersistContentInClientCache <Boolean>] [-RequiresUserInteraction <Boolean>] [-RunInstallationProgramAs32BitProcessOn64BitClient <Boolean>] [-RunScriptAs32bitProcessOn64bitClient <Boolean>] [-UninstallProgram <String>] [-UninstallStartIn <String>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: AddDeploymentTypeByWebAppInstallerAuto

Add-CMDeploymentType -ApplicationName <String> -AutoIdentifyFromInstallationFile -WebAppInstaller -WebAppURL <String> [-AdministratorComment <String>] [-DeploymentTypeName <String>] [-Language <String[]>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: AddDeploymentTypeByWindows8AppInstallerAuto

Add-CMDeploymentType -ApplicationName <String> -AutoIdentifyFromInstallationFile -ForceForUnknownPublisher <Boolean> -InstallationFileLocation <String> -Windows8AppInstaller [-AdministratorComment <String>] [-AllowClientsToUseFallbackSourceLocationForContent <Boolean>] [-DeploymentTypeName <String>] [-Language <String[]>] [-OnSlowNetworkMode {DoNothing | Download | DownloadContentForStreaming}] [-TriggerVPN <Boolean>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: AddDeploymentTypeByWinPhone8DeepLinkInstallerAuto

Add-CMDeploymentType -ApplicationName <String> -AutoIdentifyFromInstallationFile -ForceForUnknownPublisher <Boolean> -InstallationFileLocation <String> -WinPhone8DeepLinkInstaller [-AdministratorComment <String>] [-DeploymentTypeName <String>] [-Language <String[]>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: AddDeploymentTypeByWinPhone8InstallerAuto

Add-CMDeploymentType -ApplicationName <String> -AutoIdentifyFromInstallationFile -ForceForUnknownPublisher <Boolean> -InstallationFileLocation <String> -WinPhone8Installer [-AdministratorComment <String>] [-DeploymentTypeName <String>] [-Language <String[]>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: AddDeploymentTypeByWMInstallerAuto

Add-CMDeploymentType -ApplicationName <String> -AutoIdentifyFromInstallationFile -ForceForUnknownPublisher <Boolean> -InstallationFileLocation <String> -WMInstaller [-AdministratorComment <String>] [-AllowUsersToUninstallThisContent <Boolean>] [-DeploymentTypeName <String>] [-Language <String[]>] [-PfxFileLocation <String>] [-PfxFilePassword <SecureString>] [-SignContentFile <Boolean>] [-SignedContentFileLocation <String>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: AddDeploymentTypeByWMInstallerManual
Add-CMDeploymentType -ApplicationName <String> -ContentLocation <String> -DeploymentTypeName <String> -ManualSpecifyDeploymentType -WMInstaller [-AdministratorComment <String>] [-AllowUsersToUninstallThisContent <Boolean>] [-Language <String[]>] [-PfxFileLocation <String>] [-PfxFilePassword <SecureString>] [-SignContentFile <Boolean>] [-SignedContentFileLocation <String>] [-Confirm] [-WhatIf] [<CommonParameters>]

Detailed Description

The **Add-CMDeploymentType** cmdlet adds a deployment type for an application. A deployment type is contained within an application and contains the information that Microsoft System Center 2012 Configuration Manager requires to install software. A deployment type also contains rules that specify if and how the software is deployed.

Parameters

-AdministratorComment<String>

Specifies a description for the deployment type.

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

-AllowClientsToShareContentOnSameSubnet<Boolean>

Indicates whether clients can share content with other clients on the same subnet.

Aliases	none
Required?	false
Position?	named
Default Value	none

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

-AllowClientsToUseFallbackSourceLocationForContent<Boolean>

Indicates whether clients can use a fallback location provided by a management point. A fallback location point provides an alternate location for source content when the content for the deployment type is not available on any preferred distribution points.

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

-AllowUsersToUninstallThisContent<Boolean>

Indicates whether a user can uninstall the application.

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

-AndroidDeepLinkInstaller

Indicates that the deployment type configures application and deployment type information by specifying a link to the app on Google Play.

Aliases	none
Required?	true

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AndroidInstaller

Indicates that the deployment type detects application information and deployment types from an app package for Android (.apk) file.

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

-ApplicationName<String>

Specifies the name of the application that is associated with the deployment type.

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

-ApplicationNameInWindowsStore<String>

Specifies the name of the application in the Windows Store.

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

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

-AppV5xInstaller

Indicates that the deployment type detects application information and deployment types from a Microsoft Application Virtualization 5 (.appv) package file.

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

-AppvInstaller

Indicates that the deployment detects application information and deployment types from a Microsoft Application Virtualization 4 manifest (.xml) file.

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

-AutoidentifyFromInstallationFile

Indicates that the deployment type extracts information from the content file.

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

-ContentLocation<String>

Specifies the path of the content. The site system server requires permission to read the content files.

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

-DeeplinkInstaller

Indicates that the deployment type detects application information and deployment types by providing a link to the application (in the Windows Store) on a computer where the application is already installed.

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

-DeploymentTypeName<String>

Specifies the name of a deployment type.

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

-DetectDeploymentTypeByCustomScript

Indicates that the deployment type uses a custom script to detect the presence of this deployment type.

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

-EstimatedInstallationTimeMinutes<Int32>

Specifies, in minutes, the estimated installation time.

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

-ForceForUnknownPublisher<Boolean>

Indicates whether the deployment type requires file signature verification.

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

-InstallationBehaviorType<InstallationBehaviorType>

Specifies the installation behavior of the deployment type. Valid values are:

- InstallForSystem
- InstallForSystemIfResourceIsDeviceOtherwiseInstallForUser
- InstallForUser

The acceptable values for this parameter are:

InstallForSystem	
InstallForSystemIfResourceIsDeviceOtherwiseInstallForUser	
InstallForUser	

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

-InstallationFileLocation<String>

Specifies the path of the installation package.

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

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

-InstallationProgram<String>

Specifies the command line for the Windows Installer package.

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

-InstallationProgramVisibility<UserInteractionMode>

Specifies the mode in which the deployment type runs on client devices. Valid values are:

- Normal
- Minimized
- Maximized
- Hidden

The acceptable values for this parameter are:

Normal	
Minimized	
Maximized	
Hidden	

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

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

-InstallationStartIn<String>

Specifies the folder that contains the installation program for the deployment type. This folder can be an absolute path on the client, or a path to the distribution point folder that contains the installation files.

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

-iOSDeepLinkInstaller

Indicates that the deployment type configures application and deployment type information by specifying a link to the app in the App Store.

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

-iOSInstaller

Indicates that the deployment type detects application information and deployment types from an app package for iOS (.ipa) file.

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

-Language<String[]>

Specifies an array of languages that the deployment type supports.

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

-LogonRequirementType<LogonRequirementType>

Specifies the logon requirement for the deployment type. Valid values are:

- OnlyWhenNoUserLoggedIn
- OnlyWhenUserLoggedIn
- WhereOrNotUserLoggedIn

The acceptable values for this parameter are:

OnlyWhenNoUserLoggedIn	
OnlyWhenUserLoggedIn	
WhereOrNotUserLoggedIn	
WhetherOrNotUserLoggedIn	

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

-MacInstaller

Indicates that the deployment type detects application information and deployment types from a Mac OS X Installer (.cmmac) file that was created by using the CMAAppUtil tool.

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

-ManualSpecifyDeploymentType

Do not use. Configuration Manager does not currently use this parameter.

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

-MaximumAllowedRunTimeMinutes<Int32>

Specifies, in minutes, the maximum time that the program is expected to run on the client computer.

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

-MsInstaller

Indicates that the deployment type detects application information and deployment types from a Windows Installer (.msi) file.

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

-NokiaInstaller

Indicates that the deployment type detects application information and deployment types from a Nokia Symbian installation (.sis or .sisx) file.

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

-OnFastNetworkMode<OnFastNetworkMode>

Specifies the installation behavior of the deployment type on a fast network. Valid values are:

- RunFromNetwork
- RunLocal

The acceptable values for this parameter are:

RunFromNetwork	
RunLocal	

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

-OnSlowNetworkMode<ContentHandlingMode>

Specifies the installation behavior of the deployment type on a slow network. Valid values are:

- DoNothing
- Download
- DownloadContentForStreaming

The acceptable values for this parameter are:

DoNothing	
Download	
DownloadContentForStreaming	

Aliases	none
Required?	false
Position?	named
Default Value	none

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

-PersistContentInClientCache<Boolean>

Indicates whether the deployment type saves content in cache indefinitely on the client computer.

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

-PfxFileLocation<String>

Specifies the path of the Personal Information Exchange (PFX) file.

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

-PfxFilePassword<SecureString>

Specifies the password, as a secure string, for the PFX file.

Aliases	none
Required?	false
Position?	named
Default Value	none

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

-RemoteComputerName<String>

Specifies a remote computer name.

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

-RequiresUserInteraction<Boolean>

Indicates whether a user can interact with the deployment type installation to configure the installation options.

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

-RunInstallationProgramAs32BitProcessOn64BitClient<Boolean>

Indicates whether the deployment type uses Microsoft Windows-32-on-Windows-64 (WOW64) subsystem to run the installation on a 64-bit client computer.

Aliases	none
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RunScriptAs32bitProcessOn64bitClient<Boolean>

Indicates whether the deployment type uses Microsoft Windows-32-on-Windows-64 (WOW64) subsystem to run a script on a 64-bit client computer.

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

-ScriptContent<String>

Specifies the script language that you want to use to detect the deployment type.

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

-ScriptInstaller

Indicates that the deployment type uses a script to detect the presence of this deployment type.

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

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

-ScriptType<ScriptLanguage>

Specifies the script language that you want to use to detect the deployment type.

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

-SignContentFile<Boolean>

Indicates whether the deployment type requires a signed content file.

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

-SignedContentFileLocation<String>

Specifies the path of the signed content file.

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

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

-TriggerVPN<Boolean>

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

-UninstallProgram<String>

Specifies the name of the uninstall program and any parameters it requires.

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

-UninstallStartIn<String>

Specifies the folder that contains the uninstall program for the deployment type. This folder can be an absolute path on the client, or a path that is relative to the distribution point folder that contains the package.

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

-WebAppInstaller

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

-WebAppURL<String>

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

-Windows8AppInstaller

Indicates that the deployment type detects application information and deployment types from a Windows app package (.appx) file.

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

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

-WinPhone8DeeplinkInstaller

Indicates that the deployment type configures application and deployment type information by specifying a link to the app in the Windows Phone Store.

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

-WinPhone8Installer

Indicates that the deployment type detects application information and deployment types from a Windows Phone app package (.xap) file.

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

-WMIInstaller

Indicates that the deployment type detects application information and deployment types from a Windows Mobile cabinet (.cab) file.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Add an Windows Installer deployment type to an application

This command adds a Windows Installer deployment type for the application named App01d2012. The command uses the *AutoIdentifyFromIntallationFile* parameter to extract information about the deployment type from the content file, and specifies the path of the installation package. The command uses the *ForceForUnknownPublisher* parameter to specify that the deployment type verifies the signature of the content file.

```
PS C:\> Add-CMDeploymentType -MsiInstaller -ApplicationName "App01d2012" -
AutoIdentifyFromIntallationFile -InstallationFileLocation "\\CMCEN\D02\Software\RDCMan.msi"
-ForceForUnknownPublisher $True
```

Example 2: Add a deployment type that uses a script

This command adds a Windows Installer deployment type for the application named App02d2012. The command specifies the name Type01 for the deployment type. The command adds a description for the deployment type, and specifies that the deployment type supports Afrikaans and Arabic. The command uses the *InstallationProgram* to specify the command line for the Windows Installer. The command specifies that the deployment type uses a custom script to detect the presence of this deployment type. The command specifies that the script type is VBScript and specifies the script language that you will use to detect the deployment type. The command specifies that the deployment type uses Microsoft Windows-32-on-Windows-64 (WOW64) subsystem to run a script on a 64-bit client computer.

```
PS C:\> Add-CMDeploymentType -ApplicationName "App02d2012" -MsiInstaller -DeploymentTypeName
"Type01" -AdministratorComment "Div A script" -Language Afrikaans,Arabic -
InstallationProgram 'msiexec /i "\\atd-
dist01\Public\CM\DTeam\FeatureData\OSD\Tbreck\Setup1.msi"' -
DetectDeploymentTypeByCustomScript -ScriptType VBScript -ScriptContent "1231231" -
RunScriptAs32bitProcessOn64bitClient $True
```

Related topics

[Get-CMDeploymentType](#)

[Set-CMDeploymentType](#)

[Remove-CMDeploymentType](#)

Add-CMDeviceAffinityToUser

Add-CMDeviceAffinityToUser

Adds device affinity to a Configuration Manager user.

Syntax

Parameter Set: AddDeviceAffinityByUserName

```
Add-CMDeviceAffinityToUser -UserName <String[]> [-DeviceId <String> ] [-DeviceName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: AddDeviceAffinityByUserId

```
Add-CMDeviceAffinityToUser -UserId <String> [-DeviceId <String> ] [-DeviceName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Add-CMDeviceAffinityToUser** cmdlet adds device affinity to a user of Microsoft System Center 2012 Configuration Manager.

Device affinity in System Center 2012 Configuration Manager associates a user with one or more devices. Instead of deploying applications to all the user's devices, you deploy the application to the user and System Center 2012 Configuration Manager automatically installs the application on all devices that are associated with that user. Device affinity removes the need for System Center 2012 Configuration Manager to determine the names of the devices of a user before you deploy applications for that user.

For more information about user device affinity, see [How to Manage User Device Affinity in Configuration Manager](http://go.microsoft.com/fwlink/?linkid=247182) (http://go.microsoft.com/fwlink/?linkid=247182) on TechNet.

Parameters

-DeviceId<String>

Specifies a device by using an ID.

Aliases	none
Required?	false
Position?	named

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

-DeviceName<String>

Specifies a device by using a name.

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

-UserId<String>

Specifies a user by using an ID.

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

-UserName<String[]>

Specifies an array of user names to associate with the device.

Aliases	UniqueUserName
Required?	true
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Add device affinity to a user by specifying a user ID

This command adds affinity to the device named WestDivUpdates05 for the user named Patti Fuller.

```
PS C:\> Add-CMDeviceAffinityToUser -UserName "Patti Fuller" -DeviceName "WestDivUpdates05"
```

Related topics

[Approve-CMUserDeviceAffinityRequest](#)

[Deny-CMUserDeviceAffinityRequest](#)

[Get-CMUserDeviceAffinity](#)

[Get-CMUserDeviceAffinityRequest](#)

[Import-CMUserDeviceAffinity](#)

[Remove-CMDeviceAffinityFromUser](#)

Add-CMDeviceCollectionDirectMembershipRule

Add-CMDeviceCollectionDirectMembershipRule

Adds a direct membership rule to one or more Configuration Manager device collection.

Syntax

Parameter Set: ByCollectionIdAndResourceId

```
Add-CMDeviceCollectionDirectMembershipRule -CollectionId <String> -ResourceId <Int32> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByCollectionIdAndResourceValue

```
Add-CMDeviceCollectionDirectMembershipRule -CollectionId <String> -Resource <IResultObject> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByCollectionNameAndResourceId

```
Add-CMDeviceCollectionDirectMembershipRule -CollectionName <String> -ResourceId <Int32> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByCollectionNameAndResourceValue

```
Add-CMDeviceCollectionDirectMembershipRule -CollectionName <String> -Resource <IResultObject> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByCollectionValueAndResourceId

```
Add-CMDeviceCollectionDirectMembershipRule -Collection <IResultObject> -ResourceId <Int32> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByCollectionValueAndResourceValue

```
Add-CMDeviceCollectionDirectMembershipRule -Collection <IResultObject> -Resource <IResultObject> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Add-CMDeviceCollectionDirectMembershipRule** cmdlet adds a rule that adds a specific resource, such as a computer, to one or more device collection. You can specify the device collections by using their name, ID, or by specifying an object that represents the collections.

A direct rule lets you explicitly choose the members of the device collection. For more information on collection rules in Microsoft System Center 2012 Configuration Manager, see [Introduction to Collections in Configuration Manager](http://go.microsoft.com/fwlink/p/?LinkID=259433) (<http://go.microsoft.com/fwlink/p/?LinkID=259433>) on TechNet.

Parameters

-Collection<IResultObject>

Specifies a Configuration Manager device collection object. To obtain a device collection object, use the **Get-CMDeviceCollection** cmdlet.

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

-CollectionId<String>

Specifies the IDs of the device collections where the rule is applied.

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

-CollectionName<String>

Specifies the name of the device collection where the rule is applied.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

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

-Resource<IResultObject>

Specifies the type of the device that is added to the device collections.

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

-ResourceId<Int32>

Specifies the ID of a resource in a direct membership rule.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Add a direct membership rule

This command adds the direct membership rule that has the ID Res_94412512 to the device collection named All Mobile Devices.

```
PS C:\> Add-CMDeviceCollectionDirectMembershipRule -CollectionName "All Mobile Devices" -ResourceId "Res_94412512"
```

Related topics

[Get-CMDeviceCollectionDirectMembershipRule](#)

[Remove-CMDeviceCollectionDirectMembershipRule](#)

Add- CMDeviceCollectionExcludeMembershipRule

Add-CMDeviceCollectionExcludeMembershipRule

Adds an exclude membership rule to one or more Configuration Manager device collections.

Syntax

Parameter Set: ByCollectionIdAndExcludeCollectionId

```
Add-CMDeviceCollectionExcludeMembershipRule -CollectionId <String> -ExcludeCollectionId <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByCollectionIdAndExcludeCollectionName

```
Add-CMDeviceCollectionExcludeMembershipRule -CollectionId <String> -ExcludeCollectionName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByCollectionIdAndExcludeCollectionValue

```
Add-CMDeviceCollectionExcludeMembershipRule -CollectionId <String> -ExcludeCollection <IResultObject> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByCollectionNameAndExcludeCollectionId

```
Add-CMDeviceCollectionExcludeMembershipRule -CollectionName <String> -ExcludeCollectionId <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByCollectionNameAndExcludeCollectionName

```
Add-CMDeviceCollectionExcludeMembershipRule -CollectionName <String> -ExcludeCollectionName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByCollectionNameAndExcludeCollectionValue

```
Add-CMDeviceCollectionExcludeMembershipRule -CollectionName <String> -ExcludeCollection <IResultObject> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByCollectionValueAndExcludeCollectionId

```
Add-CMDeviceCollectionExcludeMembershipRule -Collection <IResultObject> -ExcludeCollectionId <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByCollectionValueAndExcludeCollectionName

```
Add-CMDeviceCollectionExcludeMembershipRule -Collection <IResultObject> -ExcludeCollectionName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByCollectionValueAndExcludeCollectionValue

```
Add-CMDeviceCollectionExcludeMembershipRule -Collection <IResultObject> -ExcludeCollection <IResultObject> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Add-CMDeviceCollectionExcludeMembershipRule** cmdlet adds a rule that excludes the members of another collection from the device collections where the rule is applied. You can specify the device collections where the rule is applied by using their names, IDs, or by specifying an object that represents the collections. You can specify the collection whose members are excluded by using its name, ID, or an object that represents the collection.

Microsoft System Center 2012 Configuration Manager dynamically updates the membership of the device collection on a schedule if the membership of the excluded collection changes. For more information on these rules, see [Introduction to Collections in Configuration Manager](http://go.microsoft.com/fwlink/p/?LinkID=259433) (<http://go.microsoft.com/fwlink/p/?LinkID=259433>) on TechNet.

Parameters

-Collection<IResultObject>

Specifies a Configuration Manager device collection object. To obtain a device collection object, use the **Get-CMDeviceCollection** cmdlet.

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

-CollectionId<String>

Specifies the IDs of the device collections where the rule is applied.

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

-CollectionName<String>

Specifies the name of the device collection where the rule is applied.

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

-ExcludeCollection<IResultObject>

Specifies an object that represents the collection whose members are excluded from the device collections.

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

-ExcludeCollectionId<String>

Specifies the ID of the collection whose members are excluded in the device collection.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

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

-ExcludeCollectionName<String>

Specifies the name of the collection whose members are excluded from the device collections.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Add an exclude collection rule to a single device collection

This command excludes the members of the All Mobile Devices collection, which has the ID SMSDM001, from the device collection, which has the ID 9990000D0.

```
PS C:\> Add-CMDeviceCollectionExcludeMembershipRule -CollectionId "9990000D" -  
ExcludeCollectionId "SMSDM001"
```

Related topics

[Get-CMDeviceCollectionExcludeMembershipRule](#)

[Remove-CMDeviceCollectionExcludeMembershipRule](#)

[Get-CMDeviceCollection](#)

Add- CMDeviceCollectionIncludeMembershipRule

Add-CMDeviceCollectionIncludeMembershipRule

Adds an include membership rule to one or more Configuration Manager device collections.

Syntax

Parameter Set: ByCollectionIdAndIncludeCollectionId

```
Add-CMDeviceCollectionIncludeMembershipRule -CollectionId <String> -IncludeCollectionId <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByCollectionIdAndIncludeCollectionName

```
Add-CMDeviceCollectionIncludeMembershipRule -CollectionId <String> -IncludeCollectionName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByCollectionIdAndIncludeCollectionValue

```
Add-CMDeviceCollectionIncludeMembershipRule -CollectionId <String> -IncludeCollection <IResultObject> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByCollectionNameAndIncludeCollectionId

```
Add-CMDeviceCollectionIncludeMembershipRule -CollectionName <String> -IncludeCollectionId <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByCollectionNameAndIncludeCollectionName

```
Add-CMDeviceCollectionIncludeMembershipRule -CollectionName <String> -IncludeCollectionName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByCollectionNameAndIncludeCollectionValue

```
Add-CMDeviceCollectionIncludeMembershipRule -CollectionName <String> -IncludeCollection <IResultObject> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByCollectionValueAndIncludeCollectionId

```
Add-CMDeviceCollectionIncludeMembershipRule -Collection <IResultObject> -IncludeCollectionId <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByCollectionValueAndIncludeCollectionName

```
Add-CMDeviceCollectionIncludeMembershipRule -Collection <IResultObject> -IncludeCollectionName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByCollectionValueAndIncludeCollectionValue

```
Add-CMDeviceCollectionIncludeMembershipRule -Collection <IResultObject> -IncludeCollection <IResultObject> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Add-CMDeviceCollectionIncludeMembershipRule** cmdlet adds a rule that includes the members of another collection in the device collections where the rule is applied. You can specify the device collections where the rule is applied by using their names, IDs, or by specifying an object that represents the collections. You can specify the collection whose members are included by using its name, ID, or an object that represents the collection.

Configuration Manager dynamically updates the membership of the device collection on a schedule if the membership of the included collection changes. For more information on these rules in Microsoft System Center 2012 Configuration Manager, see [Introduction to Collections in Configuration Manager](http://go.microsoft.com/fwlink/p/?LinkID=259433) (<http://go.microsoft.com/fwlink/p/?LinkID=259433>) on TechNet.

Parameters

-Collection<IResultObject>

Specifies a Configuration Manager device collection object. To obtain a device collection object, use the **Get-CMDeviceCollection** cmdlet.

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

-CollectionId<String>

Specifies the IDs of the device collections where the rule is applied.

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

-CollectionName<String>

Specifies the name of the device collection where the rule is applied.

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

-IncludeCollection<IResultObject>

Specifies an object that represents the collection whose members are included in the device collections.

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

-IncludeCollectionId<String>

Specifies the ID of the collection whose members are included in the device collection.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

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

-IncludeCollectionName<String>

Specifies the name of the collection whose members are included in the device collections.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Add an include membership rule to a device collection

This command adds a rule that includes the members of the collection that has the ID SMSDM001 in the device collection that has the ID 9990000D.

```
PS C:\> Add-CMDeviceCollectionIncludeMembershipRule -CollectionId "9990000D" -  
IncludeCollectionId "SMSDM001"
```

Related topics

[Get-CMDeviceCollectionIncludeMembershipRule](#)

[Remove-CMDeviceCollectionIncludeMembershipRule](#)

Add-CMDeviceCollectionQueryMembershipRule

Add-CMDeviceCollectionQueryMembershipRule

Adds a query membership rule to one or more Configuration Manager device collections.

Syntax

Parameter Set: ByCollectionId

```
Add-CMDeviceCollectionQueryMembershipRule -CollectionId <String> -QueryExpression <String> -RuleName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByCollectionName

```
Add-CMDeviceCollectionQueryMembershipRule -CollectionName <String> -QueryExpression <String> -RuleName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByCollectionValue

```
Add-CMDeviceCollectionQueryMembershipRule -Collection <IResultObject> -QueryExpression <String> -RuleName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Add-CMDeviceCollectionQueryMembershipRule** cmdlet adds a rule that adds devices to the collections based on a query. You can specify the device collections by using their names, IDs, or by specifying an object that represents the collections. The query is specified as a text string.

A query rule lets you dynamically update the members of a collection based on a query that is run on a schedule. For more information on collection rules in Microsoft System Center 2012

Configuration Manager, see [Introduction to Collections in Configuration Manager](http://go.microsoft.com/fwlink/p/?LinkID=259433) (<http://go.microsoft.com/fwlink/p/?LinkID=259433>) on TechNet.

Parameters

-Collection<IResultObject>

Specifies a Configuration Manager device collection object. To obtain a device collection object, use the **Get-CMDeviceCollection** cmdlet.

Aliases	none
Required?	true

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CollectionId<String>

Specifies the ID of the device collection where the rule is applied.

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

-CollectionName<String>

Specifies the name of the device collection where the rule is applied.

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

-QueryExpression<String>

Specifies the query expression that Configuration Manager uses to update the device collections.

Aliases	none
Required?	true

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RuleName<String>

Specifies the name for the rule.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Add a query membership rule

This command adds a membership rule named TPM Information to the device collection named Mobile Windows 7 Devices. The *QueryExpression* parameter specifies the query that defines the membership rule.

```
PS C:\> Add-CMDeviceCollectionQueryMembershipRule -CollectionName "Mobile Windows 7 Devices"
-QueryExpression "select SMS_R_System.ResourceId, SMS_R_System.ResourceType,
SMS_R_System.Name, SMS_R_System.SMSUniqueIdentifier, SMS_R_System.ResourceDomainORWorkgroup,
SMS_R_System.Client from SMS_R_System inner join SMS_G_System_TPM on
SMS_G_System_TPM.ResourceID = SMS_R_System.ResourceId" -RuleName "TPM Information"
```

Related topics

[Get-CMDeviceCollectionQueryMembershipRule](#)

[Remove-CMDeviceCollectionQueryMembershipRule](#)

[Get-CMDeviceCollection](#)

Add-CMDeviceCollectionToAdministrativeUser

Add-CMDeviceCollectionToAdministrativeUser

Associates a Configuration Manager device collection with an administrative user.

Syntax

Parameter Set: AddDeviceCollectionToAdminByName_Name

```
Add-CMDeviceCollectionToAdministrativeUser -AdministrativeUserName <String> -  
DeviceCollectionName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: AddDeviceCollectionToAdminById_Id

```
Add-CMDeviceCollectionToAdministrativeUser -AdministrativeUserId <Int32> -DeviceCollectionId  
<String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: AddDeviceCollectionToAdminById_Name

```
Add-CMDeviceCollectionToAdministrativeUser -AdministrativeUserName <String> -  
DeviceCollectionId <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: AddDeviceCollectionToAdminById_Object

```
Add-CMDeviceCollectionToAdministrativeUser -AdministrativeUser <IResultObject> -  
DeviceCollectionId <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: AddDeviceCollectionToAdminByName_Id

```
Add-CMDeviceCollectionToAdministrativeUser -AdministrativeUserId <Int32> -  
DeviceCollectionName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: AddDeviceCollectionToAdminByName_Object

```
Add-CMDeviceCollectionToAdministrativeUser -AdministrativeUser <IResultObject> -  
DeviceCollectionName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: AddDeviceCollectionToAdminByObject_Id

```
Add-CMDeviceCollectionToAdministrativeUser -AdministrativeUserId <Int32> -DeviceCollection  
<IResultObject> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: AddDeviceCollectionToAdminByObject_Name

```
Add-CMDeviceCollectionToAdministrativeUser -AdministrativeUserName <String> -  
DeviceCollection <IResultObject> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: AddDeviceCollectionToAdminByObject_Object

```
Add-CMDeviceCollectionToAdministrativeUser -AdministrativeUser <IResultObject> -  
DeviceCollection <IResultObject> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Add-CMDeviceCollectionToAdministrativeUser** cmdlet associates a device collection in Microsoft System Center 2012 Configuration Manager with an administrative user. This association gives the administrative user the ability to manage devices within the collection.

Parameters

-AdministrativeUser<IResultObject>

Specifies the administrative user by using an administrative user object. To obtain an administrative user object, use the [Get-CMAdministrativeUser](#) cmdlet.

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

-AdministrativeUserId<Int32>

Specifies the administrative user by using an ID.

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

-AdministrativeUserName<String>

Specifies the administrative user by using a username.

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

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

-DeviceCollection<IResultObject>

Specifies the device collection by using a device collection object. To obtain a device collection object, use the [Get-CMDeviceCollection](#) cmdlet.

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

-DeviceCollectionId<String>

Specifies the device collection by using an ID.

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

-DeviceCollectionName<String>

Specifies the device collection by using a name.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Associate a device collection with an administrative user by using names

This command associates the device collection named PhoneCollection05 with the administrative user who has username TSQA\PhoneAdmin.

```
PS C:\> Add-CMDeviceCollectionToAdministrativeUser -DeviceCollectionName "PhoneCollection05"
-AdministrativeUserName "TSQA\PhoneAdmin"
```

Related topics

[Remove-CMDeviceCollectionFromAdministrativeUser](#)

Add- CMDeviceCollectionToDistributionPointGroup

Add-CMDeviceCollectionToDistributionPointGroup

Associates a device collection with a distribution point group.

Syntax

Parameter Set: AddDeviceCollectionToDistributionPointGroupById_Id

```
Add-CMDeviceCollectionToDistributionPointGroup -DeviceCollectionId <String> -  
DistributionPointGroupId <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: AddDeviceCollectionToDistributionPointGroupById_Name

```
Add-CMDeviceCollectionToDistributionPointGroup -DeviceCollectionId <String> -  
DistributionPointGroupName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: AddDeviceCollectionToDistributionPointGroupById_Object

```
Add-CMDeviceCollectionToDistributionPointGroup -DeviceCollectionId <String> -  
DistributionPointGroup <IResultObject> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: AddDeviceCollectionToDistributionPointGroupByName_Id

```
Add-CMDeviceCollectionToDistributionPointGroup -DeviceCollectionName <String> -  
DistributionPointGroupId <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: AddDeviceCollectionToDistributionPointGroupByName_Name

```
Add-CMDeviceCollectionToDistributionPointGroup -DeviceCollectionName <String> -  
DistributionPointGroupName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: AddDeviceCollectionToDistributionPointGroupByName_Object

```
Add-CMDeviceCollectionToDistributionPointGroup -DeviceCollectionName <String> -  
DistributionPointGroup <IResultObject> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: AddDeviceCollectionToDistributionPointGroupByObject_Id

```
Add-CMDeviceCollectionToDistributionPointGroup -DeviceCollection <IResultObject> -  
DistributionPointGroupId <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: AddDeviceCollectionToDistributionPointGroupByObject_Name

```
Add-CMDeviceCollectionToDistributionPointGroup -DeviceCollection <IResultObject> -  
DistributionPointGroupName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: AddDeviceCollectionToDistributionPointGroupByObject_Object

```
Add-CMDeviceCollectionToDistributionPointGroup -DeviceCollection <IResultObject> -  
DistributionPointGroup <IResultObject> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Add-CMDeviceCollectionToDistributionPointGroup** cmdlet associates a device collection with a distribution point group. Distribution point groups provide a logical grouping of distribution points and collections for content distribution. A distribution point group is not limited to distribution points from a single site and can contain one or more distribution points from any site in the hierarchy. When you distribute content to a distribution point group, all distribution points that are members of the distribution point group receive the content. When you add a new distribution point to a distribution point group, it receives all content that it has received in previous distributions.

Parameters

-DeviceCollection<IResultObject>

Specifies the device collection by using a device collection object. To obtain a device collection object, use the [Get-CMDeviceCollection](#) cmdlet.

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

-DeviceCollectionId<String>

Specifies the device collection by using an ID.

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

-DeviceCollectionName<String>

Specifies the device collection by using a name.

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

-DistributionPointGroup<IResultObject>

Specifies the distribution point group by using an object that contains a distribution point group. To obtain such an object, use the [Get-CMDistributionPointGroup](#) cmdlet.

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

-DistributionPointGroupId<String>

Specifies the distribution point group by using an ID.

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

-DistributionPointGroupName<String>

Specifies the distribution point group by using a name.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Associate a device collection with a distribution point group by using names

This cmdlet associates the device collection named PhoneCollection05 with the distribution point group named DPG05ContosoWest.

```
PS C:\> Add-CMDeviceCollectionToDistributionPointGroup -DeviceCollectionName  
"PhoneCollection05" - DistributionPointGroupName "DPG05ContosoWest"
```

Related topics

[Remove-CMDeviceCollectionFromDistributionPointGroup](#)

Add-CMDistributionPoint

Add-CMDistributionPoint

Creates a distribution point.

Syntax

Parameter Set: DistributionPointWithSelfCert

```
Add-CMDistributionPoint -CertificateExpirationTimeUtc <DateTime> -MinimumFreeSpaceMB <Int32>
-SiteCode <String> -SiteSystemServerName <String> [-AllowPreStaging] [-
AllowRespondIncomingPxeRequest] [-ClientConnectionType {Internet | InternetAndIntranet |
Intranet} ] [-ComputersUsePxePassword <SecureString> ] [-ContentMonitoringPriority {High |
Highest | Low | Lowest | Medium} ] [-EnableAnonymous] [-EnableMulticast] [-EnablePullDP] [-
EnablePxeSupport] [-EnableScheduledMulticast <Boolean> ] [-EnableUnknownComputerSupport] [-
EnableValidateContent] [-EndIpAddress <String> ] [-EndUdpPort <Int32> ] [-
InstallInternetServer] [-MacAddressForRespondingPxeRequest <String[]> ] [-MinimumSessionSize
<Int32> ] [-MulticastMaximumClientCount <Int32> ] [-PrimaryContentLibraryLocation {A |
Automatic | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U |
V | W | X | Y | Z} ] [-PrimaryPackageShareLocation {A | Automatic | B | C | D | E | F | G |
H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z} ] [-
PxeServerResponseDelaySeconds <Int32> ] [-SecondaryContentLibraryLocation {A | Automatic | B
| C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y
| Z} ] [-SecondaryPackageShareLocation {A | Automatic | B | C | D | E | F | G | H | I | J |
K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z} ] [-SessionStartDelayMinutes
<Int32> ] [-SourceDistributionPoints <String[]> ] [-SourceDPRanks <Int32[]> ] [-
StartIpAddress <String> ] [-StartUdpPort <Int32> ] [-UserDeviceAffinity
{AllowWithAutomaticApproval | AllowWithManualApproval | DoNotUse} ] [-UserName <String> ] [-
ValidateContentSchedule <IResultObject> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: DistributionPointwithUserSpecifiedCert

```
Add-CMDistributionPoint -CertificatePassword <SecureString> -CertificatePath <String> -
MinimumFreeSpaceMB <Int32> -SiteCode <String> -SiteSystemServerName <String> [-
AllowPreStaging] [-AllowRespondIncomingPxeRequest] [-ClientConnectionType {Internet |
InternetAndIntranet | Intranet} ] [-ComputersUsePxePassword <SecureString> ] [-
ContentMonitoringPriority {High | Highest | Low | Lowest | Medium} ] [-EnableAnonymous] [-
EnableMulticast] [-EnablePullDP] [-EnablePxeSupport] [-EnableScheduledMulticast <Boolean> ]
[-EnableUnknownComputerSupport] [-EnableValidateContent] [-EndIpAddress <String> ] [-
EndUdpPort <Int32> ] [-InstallInternetServer] [-MacAddressForRespondingPxeRequest <String[]>
] [-MinimumSessionSize <Int32> ] [-MulticastMaximumClientCount <Int32> ] [-
PrimaryContentLibraryLocation {A | Automatic | B | C | D | E | F | G | H | I | J | K | L | M
| N | O | P | Q | R | S | T | U | V | W | X | Y | Z} ] [-PrimaryPackageShareLocation {A |
Automatic | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U |
V | W | X | Y | Z} ] [-PxeServerResponseDelaySeconds <Int32> ] [-
SecondaryContentLibraryLocation {A | Automatic | B | C | D | E | F | G | H | I | J | K | L |
M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z} ] [-SecondaryPackageShareLocation {A
| Automatic | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U
| V | W | X | Y | Z} ] [-SessionStartDelayMinutes <Int32> ] [-SourceDistributionPoints
```

```
<String[]> ] [-SourceDPRanks <Int32[]> ] [-StartIpAddress <String> ] [-StartUdpPort <Int32> ] [-UserDeviceAffinity {AllowWithAutomaticApproval | AllowWithManualApproval | DoNotUse} ] [-UserName <String> ] [-ValidateContentSchedule <IResultObject> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Add-CMDistributionPoint** cmdlet creates a distribution point on a site system server. A distribution point is a site system role that Microsoft System Center 2012 Configuration Manager uses to store files for clients to download, such as application content, software packages, software updates, operating system images, and boot images.

You must designate a site system server as a distribution point before you can make content available to client computers. You can add the distribution point site role to a new site system server or add the site role to an existing site system server.

Parameters

-AllowPreStaging

Indicates that the distribution point can pre-stage contents.

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

-AllowRespondIncomingPxeRequest

Indicates that the distribution point can respond to pre-boot execution environment (PXE) requests.

Aliases	none
Required?	false
Position?	named
Default Value	none

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

-CertificateExpirationTimeUtc<DateTime>

Specifies the date and time when the certificate expires.

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

-CertificatePassword<SecureString>

Specifies the password, as a secure string, for the public key infrastructure (PKI) client certificate for the distribution point.

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

-CertificatePath<String>

Specifies the import path for the PKI issued certificate that the distribution point uses.

Aliases	none
Required?	true
Position?	named

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

-ClientConnectionType<ClientConnectionTypes>

Specifies the client connection type. Valid values are:

- Internet
- InternetAndIntranet
- Intranet

The acceptable values for this parameter are:

Internet	
InternetAndIntranet	
Intranet	

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

-ComputersUsePxePassword<SecureString>

Specifies the password, as a secure string, for computers that use PXE.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

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

-ContentMonitoringPriority<Priority>

Specifies the content monitoring priority. Valid values are:

- High
- Highest
- Low
- Lowest
- Medium

The acceptable values for this parameter are:

High	
Highest	
Low	
Lowest	
Medium	

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

-EnableAnonymous

Indicates that the distribution point permits anonymous connections from Configuration Manager clients to the content library.

Aliases	none
Required?	false
Position?	named

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

-EnableMulticast

Indicates that multicast is enabled on this distribution point.

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

-EnablePulIDP

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

-EnablePxeSupport

Indicates that PXE is enabled on the distribution point.

When you enable PXE, Configuration Manager installs Windows Deployment Services on the server, if required. Windows Deployment Service is the service that performs the PXE boot to install operating systems. After you create the distribution point, Configuration Manager installs a provider in Windows Deployment Services that uses the PXE boot functions.

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

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

-EnableScheduledMulticast<Boolean>

Indicates whether you can configure when Configuration Manager deploys the operating system image from the distribution point.

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

-EnableUnknownComputerSupport

Indicates that support for unknown computers is enabled. Unknown computers are computers that are not managed by Configuration Manager.

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

-EnableValidateContent

Indicates that the distribution point validates the integrity of the content files in the package.

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

-EndIpAddress<String>

Specifies the ending IP address in the range of IP addresses that Configuration Manager uses to send data to the destination computers.

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

-EndUdpPort<Int32>

Specifies the ending port in the range of user datagram protocol (UDP) ports that Configuration Manager uses to send data to the destination computers.

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

-InstallInternetServer

Indicates that Configuration Manager installs and configures Internet Information Services (IIS) on the server if it is not already installed. IIS must be installed on all distribution points.

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

-MacAddressForRespondingPxeRequest<String[]>

Specifies an array of media access controller (MAC) addresses that the distribution point uses to respond to pre-boot execution environment (PXE) requests.

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

-MinimumFreeSpaceMB<Int32>

Specifies the amount of free space on a drive before Configuration Manager chooses a different drive and continues the copy process to that drive. Content files can span multiple drives.

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

-MinimumSessionSize<Int32>

Specifies how many requests must be received before Configuration Manager starts to deploy the operating system.

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

-MulticastMaximumClientCount<Int32>

Specifies the maximum number of destination computers that can download the operating system from this distribution point.

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

-PrimaryContentLibraryLocation<DriveType>

Specifies the primary content location. Configuration Manager copies content to the primary content location until the amount of free space reaches the value that you specified for the *MinimumFreeSpaceMB* parameter. Valid values are:

- Automatic.
- Drive letter from A: through Z:.

The acceptable values for this parameter are:

A	
---	--

Automatic	
B	
C	
D	
E	
F	
G	
H	
I	
J	
K	
L	
M	
N	
O	
P	
Q	
R	
S	
T	
U	
V	
W	
X	
Y	
Z	

Aliases	none
Required?	false
Position?	named
Default Value	none

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

-PrimaryPackageShareLocation<DriveType>

Specifies the primary package share location. Configuration Manager copies content to the primary package share location until the amount of free space reaches the value that you specified for the *MinimumFreeSpaceMB* parameter. Valid values are:

- Automatic.
- Drive letter from A: through Z:.

The acceptable values for this parameter are:

A	
Automatic	
B	
C	
D	
E	
F	
G	
H	
I	
J	
K	
L	
M	
N	
O	
P	
Q	
R	
S	
T	

U	
V	
W	
X	
Y	
Z	

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

-PxeServerResponseDelaySeconds<Int32>

Specifies, in seconds, how long the distribution point delays before it responds to computer requests when you are using multiple PXE-enabled distribution points. By default, the Configuration Manager PXE service point responds first to network PXE requests.

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

-SecondaryContentLibraryLocation<DriveType>

Specifies the secondary content location. Valid values are:

- Automatic.
- Drive letter from A: through Z:.

The acceptable values for this parameter are:

A	
Automatic	
B	
C	
D	
E	
F	
G	
H	
I	
J	
K	
L	
M	
N	
O	
P	
Q	
R	
S	
T	
U	
V	
W	
X	
Y	
Z	

Aliases	none
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SecondaryPackageShareLocation<DriveType>

Specifies the secondary package share location. Valid values are:

-- Automatic.

-- Drive letter from A: through Z:.

The acceptable values for this parameter are:

A	
Automatic	
B	
C	
D	
E	
F	
G	
H	
I	
J	
K	
L	
M	
N	
O	
P	
Q	
R	
S	

T	
U	
V	
W	
X	
Y	
Z	

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

-SessionStartDelayMinutes<Int32>

Specifies the number of minutes that Configuration Manager waits before it responds to the first deployment request.

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

-SiteCode<String>

Specifies the site code for the Configuration Manager site that hosts this site system role.

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

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

-SiteSystemServerName<String>

Specifies the name of a server that hosts a site system role.

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

-SourceDistributionPoints<String[]>

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

-SourceDPRanks<Int32[]>

Aliases	none
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-StartIpAddress<String>

Specifies the starting IP address in the range of IP addresses that Configuration Manager uses to send data to the destination computers.

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

-StartUdpPort<Int32>

Specifies the starting port in the range of user datagram protocol (UDP) ports that Configuration Manager uses to send data to the destination computers.

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

-UserDeviceAffinity<UserDeviceAffinityType>

Specifies how you want the distribution point to associate users with the destination computer for PXE deployments. Valid values are:

-- AllowWithAutomaticApproval

-- AllowWithManualApproval

-- DoNotUse

The acceptable values for this parameter are:

AllowWithAutomaticApproval	
AllowWithManualApproval	
DoNotUse	

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

-UserName<String>

Specifies the name of the user that the distribution site system components use to connect to the primary site database.

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

-ValidateContentSchedule<IResultObject>

Specifies a **CMSchedule** object. A **CMSchedule** object defines the schedule for validating the integrity of content files on the distribution point. To create a **CMSchedule** object, use the **New-CMSchedule** cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Add a distribution point

This command installs and configures a distribution point to the site server named CMDIV-WEST04.CORP.CONTOSO.COM that is hosted on the Configuration Manager site that has the site code CM2.

```
PS C:\> Add-CMDistributionPoint -SiteSystemServerName "CMDIV-WEST04.CORP.CONTOSO.COM" -  
SiteCode "CM2" -InstallInternetServer -SslState $True -ClientConnectionType Intranet -  
EnableAnonymous -CertificateExpirationDate "2111/11/13 17:45:00" -EnablePrestage -  
MinimumFreeSpace 500 -PrimaryLocation A -SecondaryLocation B -PrimaryShare C -SecondaryShare  
D -EnablePxe -AllowRespondToPxeRequest -EnableUnknownMachinesSupport -UserDeviceAffinity  
AllowUserDeviceAffinityWithManualApproval -PxeResponseDelay 10 -EnableMulticast -UserName  
"contoso\evan" -StartUdpPort 1000 -EndUdpPort 64000 -ClientTransferRate Profile10Mbps -  
MulticastMaximumClient 101 -EnableSchedule -SessionStartDelay 16 -MinimumSessionSize 21
```

Related topics

[New-CMSchedule](#)

[Get-CMDistributionPoint](#)

[Set-CMDistributionPoint](#)

[Update-CMDistributionPoint](#)

[Remove-CMDistributionPoint](#)

[Get-CMDistributionPointGroup](#)

Add-CMDistributionPointToGroup

Add-CMDistributionPointToGroup

Adds a distribution point to a distribution point group.

Syntax

Parameter Set: AddDistributionPointToGroupById_Id

```
Add-CMDistributionPointToGroup -DistributionPointGroupId <String> -DistributionPointId <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: AddDistributionPointToGroupById_Name

```
Add-CMDistributionPointToGroup -DistributionPointGroupName <String> -DistributionPointId <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: AddDistributionPointToGroupById_Object

```
Add-CMDistributionPointToGroup -DistributionPointGroup <IResultObject> -DistributionPointId <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: AddDistributionPointToGroupByName_Id

```
Add-CMDistributionPointToGroup -DistributionPointGroupId <String> -DistributionPointName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: AddDistributionPointToGroupByName_Name

```
Add-CMDistributionPointToGroup -DistributionPointGroupName <String> -DistributionPointName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: AddDistributionPointToGroupByName_Object

```
Add-CMDistributionPointToGroup -DistributionPointGroup <IResultObject> -DistributionPointName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: AddDistributionPointToGroupByObject_Id

```
Add-CMDistributionPointToGroup -DistributionPoint <IResultObject> -DistributionPointGroupId <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: AddDistributionPointToGroupByObject_Name

```
Add-CMDistributionPointToGroup -DistributionPoint <IResultObject> -DistributionPointGroupName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: AddDistributionPointToGroupByObject_Object

```
Add-CMDistributionPointToGroup -DistributionPoint <IResultObject> -DistributionPointGroup <IResultObject> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Add-CMDistributionPointToGroup** cmdlet adds a distribution point to a distribution point group. Distribution point groups provide a logical grouping of distribution points for content distribution.

You can add one or more distribution points from any site in the Microsoft System Center 2012 Configuration Manager hierarchy to the distribution point group. You can also add the distribution point to more than one distribution point group so that you can manage and monitor content from a central location for distribution points that span multiple sites.

Parameters

-DistributionPoint<IResultObject>

Specifies a distribution point object. To obtain a **CMDistributionPoint** object, use the [Get-CMDistributionPoint](#) cmdlet.

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

-DistributionPointGroup<IResultObject>

Specifies a distribution point group object. To obtain a **CMDistributionPointGroup** object, use the [Get-CMDistributionPointGroup](#) cmdlet.

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

-DistributionPointGroupId<String>

Specifies the ID of a distribution point group.

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

-DistributionPointGroupName<String>

Specifies the name of a distribution point group.

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

-DistributionPointId<String>

Specifies the ID of a distribution point.

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

-DistributionPointName<String>

Specifies the name of a distribution point.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Add a distribution point to a group

This command adds the distribution point that has the Id FA921CF2-89C9-407D-A21D-FE6947F2C00A to the distribution point group named DPG01.

```
PS C:\> Add-CMDistributionPointToGroup -DistributionPointGroupName "DPG01" -Id "{FA921CF2-89C9-407D-A21D-FE6947F2C00A}"
```

Related topics

[Remove-CMDistributionPointFromGroup](#)

[Get-CMDistributionPointGroup](#)

[Get-CMDistributionPoint](#)

Add-CMDriverToDriverPackage

Add-CMDriverToDriverPackage

Adds a device driver to a driver package.

Syntax

Parameter Set: AddDriverToDriverPackageById_Id

```
Add-CMDriverToDriverPackage -DriverId <String> -DriverPackageId <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: AddDriverToDriverPackageById_Name

```
Add-CMDriverToDriverPackage -DriverId <String> -DriverPackageName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: AddDriverToDriverPackageById_Object

```
Add-CMDriverToDriverPackage -DriverId <String> -DriverPackage <IResultObject> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: AddDriverToDriverPackageByName_Id

```
Add-CMDriverToDriverPackage -DriverName <String> -DriverPackageId <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: AddDriverToDriverPackageByName_Name

```
Add-CMDriverToDriverPackage -DriverName <String> -DriverPackageName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: AddDriverToDriverPackageByName_Object

```
Add-CMDriverToDriverPackage -DriverName <String> -DriverPackage <IResultObject> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: AddDriverToDriverPackageByObject_Id

```
Add-CMDriverToDriverPackage -Driver <IResultObject> -DriverPackageId <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: AddDriverToDriverPackageByObject_Name

```
Add-CMDriverToDriverPackage -Driver <IResultObject> -DriverPackageName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: AddDriverToDriverPackageByObject_Object

```
Add-CMDriverToDriverPackage -Driver <IResultObject> -DriverPackage <IResultObject> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Add-CMDriverToDriverPackage** cmdlet adds a device driver to a driver package. You can add Windows device drivers that have been imported into the driver catalog to an existing driver package. When a device driver is added to a driver package, Microsoft System Center 2012 Configuration Manager copies the device driver content from the driver source location to the driver package.

Parameters

-Driver<IResultObject>

Specifies a driver object. To obtain a **CMDriver** object, use the **Get-CMDriver** cmdlet.

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

-DriverId<String>

Specifies the ID of a driver.

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

-DriverName<String>

Specifies the name of a driver.

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

-DriverPackage<IResultObject>

Specifies a driver package object. To obtain a **CMDriverPackage** object, use the **Get-CMDriverPackage** cmdlet.

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

-DriverPackageld<String>

Specifies the ID of a driver package.

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

-DriverPackageName<String>

Specifies the name of a driver package.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Add a driver to a driver package

This command adds the driver named Adaptec Embedded SCSI HostRAID Controller to the driver package named DrvPkg01.

```
PS C:\> Add-CMDriverToDriverPackage -DriverName "Adaptec Embedded SCSI HostRAID Controller"  
-DriverPackageName "DrvPkg01"
```

Related topics

[Remove-CMDriverFromDriverPackage](#)

[Get-CMDriver](#)

[Get-CMDriverPackage](#)

Add-CMEndpointProtectionPoint

Add-CMEndpointProtectionPoint

Adds a site system role for Endpoint Protection.

Syntax

Parameter Set: EndpointProtectionP

```
Add-CMEndpointProtectionPoint -LicenseAgreed <Boolean> -ProtectionService  
{AdvancedMembership | BasicMembership | DoNotJoinMaps} -SiteCode <String> -  
SiteSystemServerName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Add-CMEndpointProtectionPoint** cmdlet adds a site system role for System Center 2012 Endpoint Protection to a Microsoft System Center 2012 Configuration Manager site.

Endpoint Protection lets you manage antimalware policies and Windows Firewall security for client computers in System Center 2012 Configuration Manager. In order to use Endpoint Protection with System Center 2012 Configuration Manager, you must install a single site system role for Endpoint Protection, either in the central site or in a stand-alone primary site. For more information about Endpoint Protection in System Center 2012 Configuration Manager, see [Endpoint Protection in Configuration Manager](http://go.microsoft.com/fwlink/?LinkId=268427) (<http://go.microsoft.com/fwlink/?LinkId=268427>) on TechNet.

Parameters

-LicenseAgreed<Boolean>

Specifies whether you agree to the Endpoint Protection software licensing terms.

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

-ProtectionService<MapsMembershipType>

Specifies the type of membership you have for Microsoft Active Protection Service (MAPS). Valid values are:

- AdvancedMembership
- BasicMembership
- DoNotJoinMaps

The acceptable values for this parameter are:

AdvancedMembership	
BasicMembership	
DoNotJoinMaps	

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

-SiteCode<String>

Specifies the site code for a Configuration Manager site.

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

-SiteSystemServerName<String>

Specifies the name of a server that hosts a site system role.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Add a site system role

This command adds an Endpoint Protection point for the site that has the site code CM1. The specified computer hosts the role. The command also specifies that you accept the terms of the license agreement and have a basic membership for Endpoint Protection.

```
PS C:\> Add-CMEndpointProtectionPoint -LicenseAgreed $True -ProtectionService  
BasicMembership -SiteCode "CM1" -SiteSystemServerName "CMEPPoint.Western.Contoso.com"
```

Related topics

[Get-CMEndpointProtectionPoint](#)

[Remove-CMEndpointProtectionPoint](#)

[Set-CMEndpointProtectionPoint](#)

Add-CMEnrollmentPoint

Add-CMEnrollmentPoint

Adds an enrollment point to Configuration Manager.

Syntax

Parameter Set: EnrollmentPoint

```
Add-CMEnrollmentPoint -SiteCode <String> -SiteSystemServerName <String> [-IISWebsite <String> ] [-PortNumber <Int32> ] [-UserName <String> ] [-WebApplicationName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Add-CMEnrollmentPoint** cmdlet adds an enrollment point to a Microsoft System Center 2012 Configuration Manager site. An enrollment point is a site system role that manages enrollment requests from mobile devices.

When System Center 2012 Configuration Manager enrolls a mobile device, it installs a System Center 2012 Configuration Manager client. The client provides management capabilities that include hardware inventory, software deployment, settings, and remote wipe. To enroll mobile devices, use Microsoft Certificate Services with an enterprise certification authority (CA). You need a System Center 2012 Configuration Manager enrollment point site system role, as well as an enrollment proxy point site system role. For more information about site system roles, see [Install and Configure Site System Roles for Configuration Manager](http://go.microsoft.com/fwlink/?LinkId=262649) (<http://go.microsoft.com/fwlink/?LinkId=262649>) on TechNet.

Parameters

-IISWebsite<String>

Specifies the Internet Information Services (IIS) website installed on the enrollment point server.

Aliases	none
Required?	false
Position?	named
Default Value	none

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

-PortNumber<Int32>

Specifies the port to use with an enrollment point.

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

-SiteCode<String>

Specifies the site code for a Configuration Manager site.

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

-SiteSystemServerName<String>

Specifies the name of a server that hosts a site system role.

Aliases	none
Required?	true
Position?	named
Default Value	none

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

-UserName<String>

Specifies an account that the enrollment point uses to connect to the Configuration Manager database.

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

-WebApplicationName<String>

Specifies the name of the web application used for enrollment.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Add an enrollment point

This command adds an enrollment point for the site that has the site code CM4 to the server named Server04.Building02.Contoso.com. The command specifies an IIS website and port number, and the user name that the enrollment point uses to connect to the System Center 2012 Configuration Manager database. This command also specifies the web application name.

```
PS C:\> Add-CMEnrollmentPoint -SiteCode "CM4" -SiteSystemServerName  
"Server04.Building02.Contoso.com" -IISWebsite "Intranet17" -PortNumber 80 -UserName  
"QADept\Admins" -WebApplicationName "Tracker"
```

Related topics

[Get-CMEnrollmentPoint](#)

[Remove-CMEnrollmentPoint](#)

[Add-CMEnrollmentProxyPoint](#)

Add-CMEnrollmentProxyPoint

Add-CMEnrollmentProxyPoint

Adds an enrollment proxy point to Configuration Manager.

Syntax

Parameter Set: EnrollmentProxyPoint

```
Add-CMEnrollmentProxyPoint -SiteCode <String> -SiteSystemServerName <String> [-IISWebsite <String> ] [-PortNumber <Int32> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Add-CMEnrollmentProxyPoint** cmdlet adds an enrollment proxy point to a Microsoft System Center 2012 Configuration Manager site. An enrollment proxy point is a site system role.

When System Center 2012 Configuration Manager enrolls a mobile device, it installs a System Center 2012 Configuration Manager client. The client provides management capabilities that include hardware inventory, software deployment, settings, and remote wipe. To enroll mobile devices, use Microsoft Certificate Services with an enterprise certification authority (CA). You need a System Center 2012 Configuration Manager enrollment proxy point site system role, as well as an enrollment point site system role. For more information about site system roles, see [Install and Configure Site System Roles for Configuration Manager](http://go.microsoft.com/fwlink/?LinkId=262649) (<http://go.microsoft.com/fwlink/?LinkId=262649>) on TechNet.

Parameters

-IISWebsite<String>

Specifies the Internet Information Services (IIS) website installed on the enrollment proxy point server.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

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

-PortNumber<Int32>

Specifies the port that client computers use to connect with an enrollment proxy point.

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

-SiteCode<String>

Specifies the site code for a Configuration Manager site.

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

-SiteSystemServerName<String>

Specifies the name of a server that hosts a site system role.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

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

Example 1: Add an enrollment proxy point

This command adds an enrollment proxy point for the System Center 2012 Configuration Manager site that has the site code CM1. The specified computer hosts the role.

```
PS C:\> Add-CMEnrollmentProxyPoint -SiteCode "CM1" -SiteSystemServerName  
"CMEnrollmentProxyPoint.Western.Contoso.com"
```

Related topics

[Get-CMEnrollmentProxyPoint](#)

[Remove-CMEnrollmentProxyPoint](#)

[Add-CMEnrollmentPoint](#)

Add-CMFallbackStatusPoint

Add-CMFallbackStatusPoint

Adds a fallback status point to a Configuration Manager site.

Syntax

Parameter Set: FallbackStatusPoint

```
Add-CMFallbackStatusPoint -SiteCode <String> -SiteSystemServerName <String> -StateMessageNum <Int32> -ThrottleInterval <Int32> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Add-CMFallbackStatusPoint** cmdlet adds a fallback status point to a Microsoft System Center 2012 Configuration Manager site. To add this site system role, specify the site code for the System Center 2012 Configuration Manager site and the name of the computer that hosts the role. You also need to specify a throttle interval and the number of messages for that throttle window.

System Center 2012 Configuration Manager can use one or more fallback status points to collect state messages for a site and send them to System Center 2012 Configuration Manager. Throttling prevents the fallback status point from sending too many messages together, which can affect performance. For more information about site system roles, see [Install and Configure Site System Roles for Configuration Manager](http://go.microsoft.com/fwlink/?LinkId=262649) (<http://go.microsoft.com/fwlink/?LinkId=262649>) on TechNet.

Parameters

-SiteCode<String>

Specifies the site code for a Configuration Manager site.

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

-SiteSystemServerName<String>

Specifies the name of a server that hosts a site system role.

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

-StateMessageNum<Int32>

Specifies the number of state messages that a fallback status point can send to Configuration Manager within a throttle interval.

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

-ThrottleInterval<Int32>

Specifies the throttle interval, in minutes, for a fallback status point. Configuration Manager processes a limited number of state messages during this period.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

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

Example 1: Add a fallback status point

This command adds a fallback status point for the site that has the site code CM1. The specified computer hosts the role. The command also specifies number of messages and throttle time period for the fallback status point.

```
PS C:\> Add-CMFallbackStatusPoint -SiteCode "CM1" -SiteSystemServerName  
"CMFSPPoint.Western.Contoso.com" -StateMessageNum 10000 -ThrottleInterval 60
```

Related topics

[Get-CMFallbackStatusPoint](#)

[Remove-CMFallbackStatusPoint](#)

[Set-CMFallbackStatusPoint](#)

Add-CMManagementPoint

Add-CMManagementPoint

Adds a management point to Configuration Manager.

Syntax

Parameter Set: MPByHttpAndDBReplica

```
Add-CMManagementPoint -DatabaseName <String> -SiteCode <String> -SiteSystemServerName <String> -SqlServerFqdnName <String> [-GenerateAlert] [-SqlServerInstanceName <String> ] [-UserName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: MPByHttpAndSiteDB

```
Add-CMManagementPoint -SiteCode <String> -SiteSystemServerName <String> [-GenerateAlert] [-UserName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: MPByHttpsAndDBReplica

```
Add-CMManagementPoint -DatabaseName <String> -EnableSsl -SiteCode <String> -SiteSystemServerName <String> -SqlServerFqdnName <String> [-AllowDevice] [-ClientConnectionType {Internet | InternetAndIntranet | Intranet} ] [-GenerateAlert] [-SqlServerInstanceName <String> ] [-UserName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: MPByHttpsAndSiteDB

```
Add-CMManagementPoint -EnableSsl -SiteCode <String> -SiteSystemServerName <String> [-AllowDevice] [-ClientConnectionType {Internet | InternetAndIntranet | Intranet} ] [-GenerateAlert] [-UserName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Add-CMManagementPoint** cmdlet adds a management point to Microsoft System Center 2012 Configuration Manager. A management point is a System Center 2012 Configuration Manager site system role that provides policy and service information to clients and receives configuration data from clients.

Parameters

-AllowDevice

Indicates that the management point supports device clients.

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

-ClientConnectionType<ClientConnectionTypes>

Specifies the type of the client connection. Valid values are:

- Internet
- InternetAndIntranet
- Intranet

The acceptable values for this parameter are:

Internet	
InternetAndIntranet	
Intranet	

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

-DatabaseName<String>

Specifies the name of the site database or site database replica that the management point uses to query for site database information.

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

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

-EnableSsl

Indicates that the cmdlet enables SSL for the management point.

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

-GenerateAlert

Indicates that Configuration Manager generates an alert when the management point is not healthy.

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

-SiteCode<String>

Specifies the site code of the Configuration Manager site that hosts the site system role.

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

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

-SiteSystemServerName<String>

Specifies the name of the server that hosts the site system role.

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

-SqlServerFqdnName<String>

Specifies a fully qualified domain name (FQDN) for Internet-based and intranet-based clients to use when they communicate with the site system. You must specify this parameter if Internet-based client systems communicate with the site system.

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

-SqlServerInstanceName<String>

Specifies the name of the SQL Server instance that clients use to communicate with the site system.

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

-UserName<String>

Specifies a domain user account that the management point uses to access site information.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Add a management point

This command adds a management point to a System Center 2012 Configuration Manager installation. The command specifies the following information about the management point:

- The new management point appears on the site system named CMDEV-TEST02.TSQA.CONTOSO.COM. This name is also the fully qualified domain name for the SQL Server instance named MSSQLServer.
- The site has code CM2.
- The management point accepts connections from internet and intranet clients and from portable devices.
- The management point has the associated database name Test.
- The management point uses the domain user account for the user named TSQA\MPAdmin.
- The cmdlet displays all messages that the addition operation generates.

```
PS C:\> Add-CMManagementPoint -SiteSystemServerName "CMDEV-TEST02.TSQA.CONTOSO.COM" -  
SiteCode "CM2" -ClientConnectionType InternetAndIntranet -AllowDevice $True -GenerateAlert -  
SQLServerFqdnName "CMDEV-TEST02.TSQA.CONTOSO.COM" -SQLServerInstanceName "MSSQLServer" -  
DatabaseName "test" -UserName "TSQA\MPAdmin" -Verbose
```

Related topics

[Get-CMManagementPoint](#)

[Remove-CMManagementPoint](#)

[Set-CMManagementPoint](#)

Add-CMOutOfBandServicePoint

Add-CMOutOfBandServicePoint

Adds an out of band service point to Configuration Manager.

Syntax

Parameter Set: OutOfBandServicePoint

```
Add-CMOutOfBandServicePoint -MaximumSendRetryCount <Int32> -MaximumThreadCount <Int32> -  
RetryIntervalMinutes <Int32> -SiteCode <String> -SiteSystemServerName <String> -  
ThreadsOffset <Int32> [-EnableCrlChecking <Boolean> ] [-Thumbprint <String> ] [-Confirm] [-  
WhatIf] [ <CommonParameters>]
```

Parameter Set: OutOfBandServicePointWithCert

```
Add-CMOutOfBandServicePoint -Certificate <X509Certificate2> -EnableCrlChecking <Boolean> -  
MaximumSendRetryCount <Int32> -RetryIntervalMinutes <Int32> -SiteCode <String> -  
SiteSystemServerName <String> -ThreadsOffset <Int32> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Add-CMOutOfBandServicePoint** cmdlet adds an out of band service point to Microsoft System Center 2012 Configuration Manager. An out of band service point is a site system role that provisions and configures Intel Active Management Technology (AMT)-based computers for Microsoft System Center 2012 Configuration Manager.

For more information about out of band management for System Center 2012 Configuration Manager see [Introduction to Out of Band Management in Configuration Manager](http://go.microsoft.com/fwlink/?linkid=252706) (http://go.microsoft.com/fwlink/?linkid=252706) on TechNet.

Parameters

-Certificate<X509Certificate2>

Specifies the trusted root certificate for out of band management.

Aliases	none
Required?	true
Position?	named

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

-EnableCrlChecking<Boolean>

Indicates whether the out-of-band service point verifies the certificate revocation list (CRL) for the provisioning certificate.

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

-MaximumSendRetryCount<Int32>

Specifies the number of retries that Configuration Manager makes if it fails to turn on an Intel Active Management Technology (AMT)-based computer.

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

-MaximumThreadCount<Int32>

Specifies the maximum number of outbound connection threads that the site system role supports.

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

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

-RetryIntervalMinutes<Int32>

Specifies the number of minutes that a service point waits between attempts to power on an AMT-based computer.

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

-SiteCode<String>

Specifies the site code for the Configuration Manager site that hosts this site system role.

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

-SiteSystemServerName<String>

Specifies a fully qualified domain name (FQDN) of the server that hosts the site system role.

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

-ThreadsOffset<Int32>

Specifies the number of minutes before a scheduled activity that Configuration Manager sends commands to turn on an AMT-based computer.

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

-Thumbprint<String>

Specifies the thumbprint of the AMT provisioning certificate.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Add an out of band service point by using a site code

This command adds an out of band service point to the System Center 2012 Configuration Manager site that has the site code named CM2 on the site system named cmcen-dist02.tsqa.contoso.com.

```
PS C:\> Add-CMOutOfBandServicePoint -SiteSystemServerName "cmcen-dist02.tsqa.contoso.com" -
SiteCode "CM2" -Thumbprint "916EC36F1068D47DE48A02A788A9DB137CD0B674"
```

Related topics

[Get-CMOutOfBandServicePoint](#)

[Remove-CMOutOfBandServicePoint](#)

[Set-CMOutOfBandServicePoint](#)



Add-CMReportingServicePoint

Add-CMReportingServicePoint

Adds a reporting service point to Configuration Manager.

Syntax

Parameter Set: ReportingServicePoint

```
Add-CMReportingServicePoint -ReportServerInstance <String> -SiteCode <String> -  
SiteSystemServerName <String> -UserName <String> [-DatabaseName <String> ] [-  
DatabaseServerName <String> ] [-FolderName <String> ] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Detailed Description

The **Add-CMReportingServicePoint** cmdlet adds a reporting service point to Configuration Manager. A reporting service point is a site system role that is installed on a server that is running Microsoft SQL Server Reporting Services.

Parameters

-DatabaseName<String>

Specifies the name of the Configuration Manager database that you want to use as the data source for reports from Microsoft SQL Server Reporting Services.

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

-DatabaseServerName<String>

Specifies the name of the Configuration Manager database server that you want to use as the data source for reports from Microsoft SQL Server Reporting Services. To specify a database instance, use the format <Server Name>\<Instance Name>.

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

-FolderName<String>

Specifies the name of the report folder on the report server.

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

-ReportServerInstance<String>

Specifies the name of an instance of SQL Server for Reporting Services.

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

-SiteCode<String>

Specifies a site code of a Configuration Manager site that hosts this site system role.

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

-SiteSystemServerName<String>

Specifies a fully qualified domain name (FQDN) of the server that hosts the site system role.

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

-UserName<String>

Specifies a user name for an account that Configuration Manager uses to connect with SQL Server Reporting Services and that gives this user access to the site database.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

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

Example 1: Add a reporting service point

This command adds a reporting service point on the computer named CMReportingServicePoint.Western.Contoso.com for the Configuration Manager site that has the site code CM1.

```
PS C:\> Add-CMReportingServicePoint -SiteCode "CM1" -SiteSystemServerName  
"CMReportingServicePoint.Western.Contoso.com"
```

Related topics

[Get-CMReportingServicePoint](#)

[Remove-CMReportingServicePoint](#)

[Set-CMReportingServicePoint](#)

Add-CMSecurityRoleToAdministrativeUser

Add-CMSecurityRoleToAdministrativeUser

Adds a security role to an administrative user or group in Configuration Manager.

Syntax

Parameter Set: AddRoleToAdminByName_Name

```
Add-CMSecurityRoleToAdministrativeUser -AdministrativeUserName <String> -RoleName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: AddRoleToAdminById_Id

```
Add-CMSecurityRoleToAdministrativeUser -AdministrativeUserId <Int32> -RoleId <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: AddRoleToAdminById_Name

```
Add-CMSecurityRoleToAdministrativeUser -AdministrativeUserName <String> -RoleId <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: AddRoleToAdminById_Object

```
Add-CMSecurityRoleToAdministrativeUser -AdministrativeUser <IResultObject> -RoleId <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: AddRoleToAdminByName_Id

```
Add-CMSecurityRoleToAdministrativeUser -AdministrativeUserId <Int32> -RoleName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: AddRoleToAdminByName_Object

```
Add-CMSecurityRoleToAdministrativeUser -AdministrativeUser <IResultObject> -RoleName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: AddRoleToAdminByObject_Id

```
Add-CMSecurityRoleToAdministrativeUser -AdministrativeUserId <Int32> -Role <IResultObject> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: AddRoleToAdminByObject_Name

```
Add-CMSecurityRoleToAdministrativeUser -AdministrativeUserName <String> -Role <IResultObject> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: AddRoleToAdminByObject_Object

```
Add-CMSecurityRoleToAdministrativeUser -AdministrativeUser <IResultObject> -Role <IResultObject> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Add-CMSecurityRoleToAdministrativeUser** cmdlet adds a security role to an administrative user or administrative group in Microsoft System Center 2012 Configuration Manager.

Permissions defined in a role represent object types and actions available for each object type. System Center 2012 Configuration Manager provides some built-in security roles. You can also create custom security roles. For more information about security roles, see [Configuring Security for Configuration Manager](http://go.microsoft.com/fwlink/?LinkID=247225) (<http://go.microsoft.com/fwlink/?LinkID=247225>) on TechNet.

You can specify an administrative user or group by name or by ID or you can use the use the **Get-CMAdministrativeUser** cmdlet to obtain a user or group object. You can specify a role to add by name or by ID or you can use the **Get-CMSecurityRole** cmdlet to obtain a role.

Parameters

-AdministrativeUser<IResultObject>

Specifies an administrative user or administrative group object. To an administrative user or administrative group object, use the **Get-AdministrativeUser** cmdlet.

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

-AdministrativeUserId<Int32>

Specifies an ID of an administrative user or administrative group.

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

-AdministrativeUserName<String>

Specifies a name of an administrative user or administrative group.

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

-Role<IResultObject>

Specifies a role object. To obtain a role object, use the **Get-CMSecurityRole** cmdlet. A role represents Configuration Manager permissions granted to a user.

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

-RoleId<String>

Specifies an ID of a role. A role represents Configuration Manager permissions granted to a user.

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

-RoleName<String>

Specifies a name of a role. A role represents Configuration Manager permissions granted to a user.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Add a named role to a named user group

This command adds a security role named SecurityRole17 to the administrative group named Western Administrators.

```
PS C:\> Add-CMSecurityRoleToAdministrativeUser -AdministrativeUserName "Western Administrators" -RoleName "SecurityRole17"
```

Example 2: Add a role to a named user group identified by using an ID

This command adds a security role that has the specified ID to the administrative group named Western Administrators.

```
PS C:\> Add-CMSecurityRoleToAdministrativeUser -AdministrativeUserName "Western Administrators" -RoleId "SMS38973"
```

Related topics

[Remove-CMSecurityRoleFromAdministrativeUser](#)

[Get-CMAdministrativeUser](#)

[Get-CMSecurityRole](#)

[Add-CMDeviceCollectionToAdministrativeUser](#)

[Add-CMSecurityScopeToAdministrativeUser](#)

[Add-CMUserCollectionToAdministrativeUser](#)

Add-CMSecurityScopeToAdministrativeUser

Add-CMSecurityScopeToAdministrativeUser

Adds a security scope to an administrative user or group in Configuration Manager.

Syntax

Parameter Set: AddScopeToAdminByName_Name

```
Add-CMSecurityScopeToAdministrativeUser -AdministrativeUserName <String> -SecurityScopeName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: AddScopeToAdminById_Id

```
Add-CMSecurityScopeToAdministrativeUser -AdministrativeUserId <Int32> -SecurityScopeId <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: AddScopeToAdminById_Name

```
Add-CMSecurityScopeToAdministrativeUser -AdministrativeUserName <String> -SecurityScopeId <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: AddScopeToAdminById_Object

```
Add-CMSecurityScopeToAdministrativeUser -AdministrativeUser <IResultObject> -SecurityScopeId <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: AddScopeToAdminByName_Id

```
Add-CMSecurityScopeToAdministrativeUser -AdministrativeUserId <Int32> -SecurityScopeName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: AddScopeToAdminByName_Object

```
Add-CMSecurityScopeToAdministrativeUser -AdministrativeUser <IResultObject> -SecurityScopeName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: AddScopeToAdminByObject_Id

```
Add-CMSecurityScopeToAdministrativeUser -AdministrativeUserId <Int32> -SecurityScope <IResultObject> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: AddScopeToAdminByObject_Name

```
Add-CMSecurityScopeToAdministrativeUser -AdministrativeUserName <String> -SecurityScope <IResultObject> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: AddScopeToAdminByObject_Object

```
Add-CMSecurityScopeToAdministrativeUser -AdministrativeUser <IResultObject> -SecurityScope <IResultObject> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Add-CMSecurityScopeToAdministrativeUser** cmdlet adds a security scope to an administrative user or administrative group in Microsoft System Center 2012 Configuration Manager.

For more information about security scopes, see [Configuring Security for Configuration Manager](http://go.microsoft.com/fwlink/?LinkID=247225) (<http://go.microsoft.com/fwlink/?LinkID=247225>) on TechNet.

You can specify an administrative user or group by name or by ID or you can use the use the **Get-CMAdministrativeUser** cmdlet to obtain a user or group object. You can specify a security scope to add by name or by ID or you can use the **Get-CMSecurityScope** cmdlet to obtain a security scope.

Parameters

-AdministrativeUser<IResultObject>

Specifies an administrative user or administrative group object. To get an administrative user or administrative group object, use the **Get-AdministrativeUser** cmdlet.

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

-AdministrativeUserId<Int32>

Specifies an ID of an administrative user or administrative group.

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

-AdministrativeUserName<String>

Specifies a name of an administrative user or administrative group.

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

-SecurityScope<IResultObject>

Specifies a security scope object. To obtain a security scope object, use the **Get-CMSecurityScope** cmdlet.

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

-SecurityScopeld<String>

Specifies the ID of a security scope.

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

-SecurityScopeName<String>

Specifies the name of a security scope. A security scope name can be Default or the name of a custom security scope.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Add a named security scope to a named administrative group

This command adds a security scope named Scope22 to an administrative group named Western Administrators.

```
PS C:\> Add-CMSecurityScopeToAdministrativeUser -AdministrativeUserName "Western Administrators" -SecurityScopeName "Scope22"
```

Example 2: Add a security scope to an administrative group by using an ID

This command adds the security scope that has the ID SMS00067 to the administrative user that has the ID 345.

```
PS C:\> Add-CMSecurityScopeToAdministrativeUser -AdministrativeUserId 345 -SecurityScopeId "SMS00067"
```

Related topics

[Remove-CMSecurityScopeFromAdministrativeUser](#)

[Get-CMAdministrativeUser](#)

[Get-CMSecurityScope](#)

[Add-CMDeviceCollectionToAdministrativeUser](#)

[Add-CMSecurityRoleToAdministrativeUser](#)

[Add-CMUserCollectionToAdministrativeUser](#)

Add-CMSoftwareUpdatePoint

Add-CMSoftwareUpdatePoint

Adds a software update point for Configuration Manager.

Syntax

Parameter Set: SumPWithWsus

```
Add-CMSoftwareUpdatePoint -SiteCode <String> -SiteSystemServerName <String> [-AnonymousWsusAccess] [-ClientConnectionType {Internet | InternetAndIntranet | Intranet}] [-UseProxy <Boolean>] [-UseProxyForAutoDeploymentRule <Boolean>] [-WsusIisPort <Int32>] [-WsusIisSslPort <Int32>] [-WsusSsl <Boolean>] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Add-CMSoftwareUpdatePoint** cmdlet adds a software update point that hosts the Windows Software Update Services (WSUS) processes. A software update point in Microsoft System Center 2012 Configuration Manager manages the transfer of information from WSUS.

Parameters

-AnonymousWsusAccess

Indicates that the software update point allows anonymous WSUS access.

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

-ClientConnectionType<ClientConnectionTypes>

Specifies the type of the client connection. Valid values are:

- Internet
- InternetAndIntranet
- Intranet

The acceptable values for this parameter are:

Internet	
InternetAndIntranet	
Intranet	

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

-SiteCode<String>

Specifies a site code for the Configuration Manager site that manages the system role for the software update point.

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

-SiteSystemServerName<String>

Specifies the name of a computer that hosts the software update point site system role.

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

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

-UseProxy<Boolean>

Indicates whether a software update point can use a proxy.

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

-UseProxyForAutoDeploymentRule<Boolean>

Indicates whether an auto deployment rule can use a proxy.

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

-WsusiisPort<Int32>

Specifies a port to use for unsecured access to the WSUS server.

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

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

-WsusiSslPort<Int32>

Specifies a port to user for secured access to the WSUS server.

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

-WsusSsl<Boolean>

Specifies whether the software update point uses SSL to connect to the WSUS server.

Aliases	SslWsus
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Add a software update point

This command adds a software update point on the computer named CMSoftwareUpdatePoint.Western.Contoso.com for the Configuration Manager site that has the site code CM1.

```
PS C:\> Add-CMSoftwareUpdatePoint -SiteCode "CM1" -SiteSystemServerName  
"CMSoftwareUpdatePoint.Western.Contoso.com"
```

Related topics

[Get-CMSoftwareUpdatePoint](#)

[Remove-CMSoftwareUpdatePoint](#)

[Set-CMSoftwareUpdatePoint](#)

[Get-CMSoftwareUpdatePointComponent](#)

Add-CMSoftwareUpdateToGroup

Add-CMSoftwareUpdateToGroup

Adds a software update to a software update group in Configuration Manager.

Syntax

Parameter Set: AddSoftwareUpdateToGroupById_Id

```
Add-CMSoftwareUpdateToGroup -SoftwareUpdateGroupId <String> -SoftwareUpdateId <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: AddSoftwareUpdateToGroupById_Name

```
Add-CMSoftwareUpdateToGroup -SoftwareUpdateGroupName <String> -SoftwareUpdateId <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: AddSoftwareUpdateToGroupById_Object

```
Add-CMSoftwareUpdateToGroup -SoftwareUpdateGroup <IResultObject> -SoftwareUpdateId <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: AddSoftwareUpdateToGroupByName_Id

```
Add-CMSoftwareUpdateToGroup -SoftwareUpdateGroupId <String> -SoftwareUpdateName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: AddSoftwareUpdateToGroupByName_Name

```
Add-CMSoftwareUpdateToGroup -SoftwareUpdateGroupName <String> -SoftwareUpdateName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: AddSoftwareUpdateToGroupByName_Object

```
Add-CMSoftwareUpdateToGroup -SoftwareUpdateGroup <IResultObject> -SoftwareUpdateName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: AddSoftwareUpdateToGroupByObject_Id

```
Add-CMSoftwareUpdateToGroup -SoftwareUpdate <IResultObject> -SoftwareUpdateGroupId <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: AddSoftwareUpdateToGroupByObject_Name

```
Add-CMSoftwareUpdateToGroup -SoftwareUpdate <IResultObject> -SoftwareUpdateGroupName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: AddSoftwareUpdateToGroupByObject_Object

```
Add-CMSoftwareUpdateToGroup -SoftwareUpdate <IResultObject> -SoftwareUpdateGroup <IResultObject> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Add-CMSoftwareUpdateToGroup** cmdlet adds a software update to a software update group in Microsoft System Center 2012 Configuration Manager. You can specify a software update by name or by ID or use the **Get-CMSoftwareUpdate** cmdlet to obtain an update. Likewise, you can specify a software update group by name or by ID or use the **Get-CMSoftwareUpdateGroup** cmdlet to obtain one.

Parameters

-SoftwareUpdate<IResultObject>

Specifies a software update object. To obtain a software update object, use the **Get-CMSoftwareUpdate** cmdlet.

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

-SoftwareUpdateGroup<IResultObject>

Specifies a software update group object. To obtain a software update group object, use the **Get-CMSoftwareUpdateGroup** cmdlet.

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

-SoftwareUpdateGroupId<String>

Specifies an ID of a software group.

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

-SoftwareUpdateGroupName<String>

Specifies a name of a software group.

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

-SoftwareUpdateId<String>

Specifies an ID of a software update.

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

-SoftwareUpdateName<String>

Specifies a name of a software update.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Add an update to a software group

This command adds a software update with the ID SMS00078 to the update group named Accounting Group updates.

```
PS C:\> Add-CMSoftwareUpdateToGroup -SoftwareUpdateGroupName "Accounting Group updates" -SoftwareUpdateId "SMS00078"
```

Example 2: Add an update to a software group by using IDs

This command adds a software update that has the ID SMS00078 to the update group with the specified ID.

```
PS C:\> Add-CMSoftwareUpdateToGroup -SoftwareUpdateGroupId "SUP00045" -SoftwareUpdateId "SMS00078"
```

Related topics

[Get-CMSoftwareUpdate](#)

[Get-CMSoftwareUpdateGroup](#)

Add-CMStateMigrationPoint

Add-CMStateMigrationPoint

Adds a state migration point in Configuration Manager.

Syntax

Parameter Set: StateMigrationPointWithDeletionAfter

```
Add-CMStateMigrationPoint -AllowFallbackSourceLocationForContent <Boolean> -  
EnableRestoreOnlyMode <Boolean> -SiteCode <String> -SiteSystemServerName <String> -  
StorageFolders <StorageDirectoryData[]> -TimeDeleteAfter <Int32> -TimeUnit {Days | Hours} [-  
BoundaryGroupName <String[]> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: StateMigrationPointWithImmediation

```
Add-CMStateMigrationPoint -AllowFallbackSourceLocationForContent <Boolean> -  
EnableRestoreOnlyMode <Boolean> -SiteCode <String> -SiteSystemServerName <String> -  
StorageFolders <StorageDirectoryData[]> [-BoundaryGroupName <String[]> ] [-  
DeleteImmediately] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Add-CMStateMigrationPoint** cmdlet adds a state migration point in Microsoft System Center 2012 Configuration Manager. A state migration point is a site system role that manages data transfer from client computers during an operating system installation process.

Parameters

-AllowFallbackSourceLocationForContent<Boolean>

Indicates whether a fallback source location is available.

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

-BoundaryGroupName<String[]>

Specifies an array of names of boundary groups. You can get a boundary group name by using the **Get-CMBoundaryGroup** cmdlet.

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

-DeleteImmediately

Indicates that Configuration Manager deletes client data immediately after the target computer downloads the data.

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

-EnableRestoreOnlyMode<Boolean>

Indicates whether to enable restore only mode. If this mode is enabled, Configuration Manager refuses new requests to store client data.

Aliases	none
Required?	true
Position?	named
Default Value	none

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

-SiteCode<String>

Specifies the Configuration Manager site that hosts this site system role.

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

-SiteSystemServerName<String>

Specifies the name of the site system server in Configuration Manager.

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

-StorageFolders<StorageDirectoryData[]>

Specifies an array of storage folders.

Aliases	none
Required?	true
Position?	named
Default Value	none

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

-TimeDeleteAfter<Int32>

Specifies a time interval to wait before client data is deleted.

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

-TimeUnit<IntervalType>

Specifies the unit of time for the *TimeDeleteAfter* parameter. Valid values are: Days and Hours.

The acceptable values for this parameter are:

Days	
Hours	

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Add a state migration point

This example specifies new storage folders for the data and adds a new migration state point by using the new storage folders.

The first command creates new a storage folder on the C: drive with a maximum number of clients setting and a minimum free space setting. The command stores the result in the \$s1 variable.

The second command creates new a storage folder on the D: drive with a maximum number of clients setting and a minimum free space setting. The command stores the result in the \$s2 variable.

The third command adds a state migration point.

```
PS C:\> $s1 = New-CMStoragefolder -StorageFolderName "C:\Sto-1" -MaximumClientNumber 100 -
MinimumFreeSpace 100 -SpaceUnit Megabyte
PS C:\> $s2 = New-CMStoragefolder -StorageFolderName "D:\Sto-2" -MaximumClientNumber 100 -
MinimumFreeSpace 10 -SpaceUnit Gigabyte
PS C:\> Add-CMStateMigrationPoint -SiteSystemServerName "Contoso-Migration.Contoso.com" -
```

```
SiteCode "CM2" -StorageFolders $s1,$s2 -DeleteImmediately -EnableRestoreOnlyMode $False -  
AllowFallbackSourceLocationForContent $False -BoundaryGroupName "CMC"
```

Related topics

[Get-CMStateMigrationPoint](#)

[Remove-CMStateMigrationPoint](#)

[Set-CMStateMigrationPoint](#)

[Get-CMBoundaryGroup](#)

Add-CMSystemHealthValidatorPoint

Add-CMSystemHealthValidatorPoint

Adds a validator point for system health in Configuration Manager.

Syntax

Parameter Set: SystemHealthValidatorPoint

```
Add-CMSystemHealthValidatorPoint -SiteCode <String> -SiteSystemServerName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Add-CMSystemHealthValidatorPoint** cmdlet adds a validator point for system health in Microsoft System Center 2012 Configuration Manager.

Parameters

-SiteCode<String>

Specifies a site code that identifies the Configuration Manager site that hosts this site system role.

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

-SiteSystemServerName<String>

Specifies the host name for a system health validator point.

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

Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Add a system health validator point to a server for a site

This command adds a system health validator point to the server named Test01.Western.Contoso.com for the site that has the site code CM4.

```
PS C:\> Add-CMSystemHealthValidatorPoint -SiteCode "CM4" -SiteSystemServerName  
"Test01.Western.Contoso.com"
```

Related topics

[Get-CMSystemHealthValidatorPoint](#)

[Remove-CMSystemHealthValidatorPoint](#)

Add-CMUserAffinityToDevice

Add-CMUserAffinityToDevice

Adds a primary user to one or more devices in the Configuration Manager hierarchy.

Syntax

Parameter Set: AddUserAffinityByDeviceName

```
Add-CMUserAffinityToDevice -DeviceName <String[]> [-UserId <String> ] [-UserName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: AddUserAffinityByDeviceId

```
Add-CMUserAffinityToDevice -DeviceId <String[]> [-UserId <String> ] [-UserName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Add-CMUserAffinityToDevice** cmdlet associates the devices with a primary user in Microsoft System Center 2012 Configuration Manager. You can specify the devices to associate with the primary user by their names or IDs. You can specify the primary user by their name or ID.

User device affinity is a method of associating a user with one or more specified devices. This allows you to deploy applications to a user without the requirement to know the name of all the user's devices. Instead of deploying the application to all the devices of a user, you deploy the application to user and the application automatically installs on all devices that are associated with that user.

Parameters

-DeviceId<String[]>

Specifies an array of IDs of the devices that you want to associate with the primary user.

Aliases	ResourceId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

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

-DeviceName<String[]>

Specifies an array of names of the devices that you want to associate with the primary user.

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

-UserId<String>

Specifies the ID of a user.

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

-UserName<String>

Specifies the name of the primary user.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

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

Example 1: Add a primary user to a device

This command adds the primary user that has the ID 2063597981 to the device named CMCEN-DIST-02.

```
PS C:\> Add-CMUserAffinityToDevice -DeviceName "CMCEN-DIST-02" -UserId "2063597981"
```

Related topics

[Remove-CMUserAffinityFromDevice](#)

Add-CMUserCollectionDirectMembershipRule

Add-CMUserCollectionDirectMembershipRule

Adds a direct membership rule to one or more Configuration Manager user collections.

Syntax

Parameter Set: ByCollectionIdAndResourceId

```
Add-CMUserCollectionDirectMembershipRule -CollectionId <String> -ResourceId <Int32> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByCollectionIdAndResourceValue

```
Add-CMUserCollectionDirectMembershipRule -CollectionId <String> -Resource <IResultObject> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByCollectionNameAndResourceId

```
Add-CMUserCollectionDirectMembershipRule -CollectionName <String> -ResourceId <Int32> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByCollectionNameAndResourceValue

```
Add-CMUserCollectionDirectMembershipRule -CollectionName <String> -Resource <IResultObject> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByCollectionValueAndResourceId

```
Add-CMUserCollectionDirectMembershipRule -Collection <IResultObject> -ResourceId <Int32> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByCollectionValueAndResourceValue

```
Add-CMUserCollectionDirectMembershipRule -Collection <IResultObject> -Resource <IResultObject> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Add-CMUserCollectionDirectMembershipRule** cmdlet adds a rule that adds a specific resource to the user collections. You can specify the user collections by using their names, IDs, or by specifying an object that represents the collections.

A direct rule lets you explicitly choose the members of the user collection. For more information on collection rules in Microsoft System Center 2012 Configuration Manager, see [Introduction to Collections in Configuration Manager](http://go.microsoft.com/fwlink/p/?LinkID=259433) (<http://go.microsoft.com/fwlink/p/?LinkID=259433>) on TechNet.

Parameters

-Collection<IResultObject>

Specifies a Configuration Manager user collection object where the rule is applied. To obtain a user collection object, use the **Get-CMUserCollection** cmdlet.

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

-CollectionId<String>

Specifies the ID of the user collection where the rule is applied.

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

-CollectionName<String>

Specifies the name of a user collection where the rule is applied.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

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

-Resource<IResultObject>

Specifies the type of the resource that the cmdlet adds to the user collections.

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

-ResourceId<Int32>

Specifies the ID of a resource that the cmdlet adds to the user collections.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Add a direct membership rule to a user collection

This command adds a direct membership rule that has the ID Res_94412512 to the Configuration Manager user collection named All Mobile Devices.

```
PS C:\> Add-CMUserCollectionDirectMembershipRule -CollectionName "All Mobile Devices" -ResourceId "Res_94412512"
```

Related topics

[Get-CMUserCollectionDirectMembershipRule](#)

[Remove-CMUserCollectionDirectMembershipRule](#)

[Get-CMUserCollection](#)

Add-CMUserCollectionExcludeMembershipRule

Add-CMUserCollectionExcludeMembershipRule

Adds an exclude membership rule to one or more Configuration Manager user collections.

Syntax

Parameter Set: ByCollectionIdAndExcludeCollectionId

```
Add-CMUserCollectionExcludeMembershipRule -CollectionId <String> -ExcludeCollectionId <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByCollectionIdAndExcludeCollectionName

```
Add-CMUserCollectionExcludeMembershipRule -CollectionId <String> -ExcludeCollectionName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByCollectionIdAndExcludeCollectionValue

```
Add-CMUserCollectionExcludeMembershipRule -CollectionId <String> -ExcludeCollection <IResultObject> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByCollectionNameAndExcludeCollectionId

```
Add-CMUserCollectionExcludeMembershipRule -CollectionName <String> -ExcludeCollectionId <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByCollectionNameAndExcludeCollectionName

```
Add-CMUserCollectionExcludeMembershipRule -CollectionName <String> -ExcludeCollectionName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByCollectionNameAndExcludeCollectionValue

```
Add-CMUserCollectionExcludeMembershipRule -CollectionName <String> -ExcludeCollection <IResultObject> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByCollectionValueAndExcludeCollectionId

```
Add-CMUserCollectionExcludeMembershipRule -Collection <IResultObject> -ExcludeCollectionId <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByCollectionValueAndExcludeCollectionName

```
Add-CMUserCollectionExcludeMembershipRule -Collection <IResultObject> -ExcludeCollectionName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByCollectionValueAndExcludeCollectionValue

```
Add-CMUserCollectionExcludeMembershipRule -Collection <IResultObject> -ExcludeCollection <IResultObject> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Add-CMUserCollectionExcludeMembershipRule** cmdlet adds a rule that excludes the members of another collection from the user collections where the rule is applied. You can specify the user collections where the rule is applied by using their names, IDs, or by specifying an object that represents the collections. You can specify the collection whose members are excluded by using its name, ID, or an object that represents the collection.

Microsoft System Center 2012 Configuration Manager dynamically updates the membership of the user collection on a schedule if the membership of the excluded collection changes. For more information on these rules, see [Introduction to Collections in Configuration Manager](http://go.microsoft.com/fwlink/p/?LinkID=259433) (<http://go.microsoft.com/fwlink/p/?LinkID=259433>) on TechNet.

Parameters

-Collection<IResultObject>

Specifies a Configuration Manager user collection object. To obtain a user collection object, use the **Get-CMUserCollection** cmdlet.

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

-CollectionId<String>

Specifies the IDs of the user collections where the rule is applied.

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

-CollectionName<String>

Specifies the name of a user collection where the rule is applied.

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

-ExcludeCollection<IResultObject>

Specifies an object that represents the collection whose members are excluded from the user collections.

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

-ExcludeCollectionId<String>

Specifies the ID of the collection whose members are excluded from the user collections.

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

-ExcludeCollectionName<String>

Specifies the name of the collection whose members are excluded from the user collections.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Add an exclude membership rule to a user collection

This command adds an exclude membership rule that has the ID SMSDM001 to the Configuration Manager user collection that has the ID 9990000D.

```
PS C:\> Add-CMUserCollectionExcludeMembershipRule -CollectionId "9990000D" -  
ExcludeCollectionId "SMSDM001"
```

Related topics

[Get-CMUserCollectionExcludeMembershipRule](#)

[Remove-CMUserCollectionExcludeMembershipRule](#)

[Get-CMUserCollection](#)

Add-CMUserCollectionIncludeMembershipRule

Add-CMUserCollectionIncludeMembershipRule

Adds an include membership rule to one or more Configuration Manager user collections.

Syntax

Parameter Set: ByCollectionIdAndIncludeCollectionId

```
Add-CMUserCollectionIncludeMembershipRule -CollectionId <String> -IncludeCollectionId <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByCollectionIdAndIncludeCollectionName

```
Add-CMUserCollectionIncludeMembershipRule -CollectionId <String> -IncludeCollectionName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByCollectionIdAndIncludeCollectionValue

```
Add-CMUserCollectionIncludeMembershipRule -CollectionId <String> -IncludeCollection <IResultObject> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByCollectionNameAndIncludeCollectionId

```
Add-CMUserCollectionIncludeMembershipRule -CollectionName <String> -IncludeCollectionId <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByCollectionNameAndIncludeCollectionName

```
Add-CMUserCollectionIncludeMembershipRule -CollectionName <String> -IncludeCollectionName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByCollectionNameAndIncludeCollectionValue

```
Add-CMUserCollectionIncludeMembershipRule -CollectionName <String> -IncludeCollection <IResultObject> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByCollectionValueAndIncludeCollectionId

```
Add-CMUserCollectionIncludeMembershipRule -Collection <IResultObject> -IncludeCollectionId <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByCollectionValueAndIncludeCollectionName

```
Add-CMUserCollectionIncludeMembershipRule -Collection <IResultObject> -IncludeCollectionName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByCollectionValueAndIncludeCollectionValue

```
Add-CMUserCollectionIncludeMembershipRule -Collection <IResultObject> -IncludeCollection <IResultObject> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Add-CMUserCollectionIncludeMembershipRule** cmdlet adds a rule that includes the members of another collection in the user collections where the rule is applied. You can specify the user collections where the rule is applied by using their names, IDs, or by specifying an object that represents the collections. You can specify the collection whose members are included by using its name, ID, or an object that represents the collection.

Microsoft System Center 2012 Configuration Manager dynamically updates the membership of the user collection on a schedule if the membership of the included collection changes. For more information on these rules, see [Introduction to Collections in Configuration Manager](http://go.microsoft.com/fwlink/p/?LinkID=259433) (<http://go.microsoft.com/fwlink/p/?LinkID=259433>) on TechNet.

Parameters

-Collection<IResultObject>

Specifies a Configuration Manager user collection object where the rule is applied. To obtain a user collection object, use the **Get-CMUserCollection** cmdlet.

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

-CollectionId<String>

Specifies the ID of the user collection where the rule is applied.

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

-CollectionName<String>

Specifies the name of the user collection where the rule is applied.

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

-IncludeCollection<IResultObject>

Specifies an object that represents the collection whose members are included in the user collections.

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

-IncludeCollectionId<String>

Specifies the ID for the collection whose members are included in the user collections.

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

-IncludeCollectionName<String>

Specifies the name of the collection whose members are included in the user collections.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Add an include membership rule to a user collection

This command adds an include membership rule that has the ID SMSDM001 to the Configuration Manager user collection that has the ID 9990000D.

```
PS C:\> Add-CMUserCollectionIncludeMembershipRule -CollectionId "9990000D" -  
IncludeCollectionId "SMSDM001"
```

Related topics

[Get-CMUserCollection](#)

[Get-CMUserCollectionIncludeMembershipRule](#)

[Remove-CMUserCollectionIncludeMembershipRule](#)

Add-CMUserCollectionQueryMembershipRule

Add-CMUserCollectionQueryMembershipRule

Adds a query membership rule to one or more Configuration Manager user collections.

Syntax

Parameter Set: ByCollectionId

```
Add-CMUserCollectionQueryMembershipRule -CollectionId <String> -QueryExpression <String> -RuleName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByCollectionName

```
Add-CMUserCollectionQueryMembershipRule -CollectionName <String> -QueryExpression <String> -RuleName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByCollectionValue

```
Add-CMUserCollectionQueryMembershipRule -Collection <IResultObject> -QueryExpression <String> -RuleName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Add-CMUserCollectionQueryMembershipRule** cmdlet adds a rule that adds users to the collections based on a query in Microsoft System Center 2012 Configuration Manager. You can specify the user collections by using their names, IDs, or by specifying an object that represents the collections. The query is specified as a text string.

A query rule lets you dynamically update the members of a collection based on a query that is run on a schedule. For more information on collection rules, see [Introduction to Collections in Configuration Manager](http://go.microsoft.com/fwlink/p/?LinkID=259433) (<http://go.microsoft.com/fwlink/p/?LinkID=259433>) on TechNet.

Parameters

-Collection<IResultObject>

Specifies a Configuration Manager user collection object. To obtain a user collection object, use the **Get-CMUserCollection** cmdlet.

Aliases	none
Required?	true

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CollectionId<String>

Specifies the ID of the user collection where the rule is applied.

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

-CollectionName<String>

Specifies the name of the user collection where the rule is applied.

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

-QueryExpression<String>

Specifies the query expression that Configuration Manager uses to update the user collections.

Aliases	none
Required?	true

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RuleName<String>

Specifies the name for the rule.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Add a rule to a collection by using a name

This command adds a rule named Remote Users by Domain to the collection named Remote Users. The command specifies the query as a string.

```
PS C:\> Add-CMUserCollectionQueryMembershipRule -CollectionName "Remote Users" -
QueryExpression "select
SMS_R_USER.ResourceID,SMS_R_USER.ResourceType,SMS_R_USER.Name,SMS_R_USER.UniqueUserName,SMS_
R_USER.WindowsNTDomain from SMS_R_User" -RuleName "Remote Users by Domain"
```

Related topics

[Get-CMUserCollection](#)

[Get-CMUserCollectionQueryMembershipRule](#)

[Remove-CMUserCollectionQueryMembershipRule](#)

Add-CMUserCollectionToAdministrativeUser

Add-CMUserCollectionToAdministrativeUser

Adds a user collection to an administrative user in Configuration Manager.

Syntax

Parameter Set: AddUserCollectionToAdminByName_Name

```
Add-CMUserCollectionToAdministrativeUser -AdministrativeUserName <String> -  
UserCollectionName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: AddUserCollectionToAdminById_Id

```
Add-CMUserCollectionToAdministrativeUser -AdministrativeUserId <Int32> -UserCollectionId  
<String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: AddUserCollectionToAdminById_Name

```
Add-CMUserCollectionToAdministrativeUser -AdministrativeUserName <String> -UserCollectionId  
<String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: AddUserCollectionToAdminById_Object

```
Add-CMUserCollectionToAdministrativeUser -AdministrativeUser <IResultObject> -  
UserCollectionId <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: AddUserCollectionToAdminByName_Id

```
Add-CMUserCollectionToAdministrativeUser -AdministrativeUserId <Int32> -UserCollectionName  
<String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: AddUserCollectionToAdminByName_Object

```
Add-CMUserCollectionToAdministrativeUser -AdministrativeUser <IResultObject> -  
UserCollectionName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: AddUserCollectionToAdminByObject_Id

```
Add-CMUserCollectionToAdministrativeUser -AdministrativeUserId <Int32> -UserCollection  
<IResultObject> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: AddUserCollectionToAdminByObject_Name

```
Add-CMUserCollectionToAdministrativeUser -AdministrativeUserName <String> -UserCollection  
<IResultObject> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: AddUserCollectionToAdminByObject_Object

```
Add-CMUserCollectionToAdministrativeUser -AdministrativeUser <IResultObject> -UserCollection  
<IResultObject> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Add-CMUserCollectionToAdministrativeUser** cmdlet adds a user collection to an administrative user. In Microsoft System Center 2012 Configuration Manager, collections represent logical groupings of users or devices. You use collections to perform tasks such as managing settings or installing software updates.

Parameters

-AdministrativeUser<IResultObject>

Specifies a **CMUserCollection** object. To obtain a user collection object, use the **Get-CMUserCollection** cmdlet.

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

-AdministrativeUserId<Int32>

Specifies a unique ID of an administrative group or user.

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

-AdministrativeUserName<String>

Specifies a name of an administrative group or user.

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

-UserCollection<IResultObject>

Specifies a **CMUserCollection** object. To obtain a user collection object, use the **Get-
CMUserCollection** cmdlet.

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

-UserCollectionId<String>

Specifies a unique ID of a user collection associated with an administrative user.

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

-UserCollectionName<String>

Specifies a name of a user collection associated with an administrative user.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Add a collection to an administrative user

This command adds a user collection to an administrative user.

```
PS C:\> Add-CMUserCollectionToAdministrativeUser -UserCollectionName "coll01" -  
AdministrativeUserName "Western\pattifuller"
```

Related topics

[Remove-CMUserCollectionFromAdministrativeUser](#)

Add- CMUserCollectionToDistributionPointGroup

Add-CMUserCollectionToDistributionPointGroup

Associates members of a user collection to a distribution point group.

Syntax

Parameter Set: AddUserCollectionToDistributionPointGroupById_Id

```
Add-CMUserCollectionToDistributionPointGroup -DistributionPointGroupId <String> -  
UserCollectionId <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: AddUserCollectionToDistributionPointGroupById_Name

```
Add-CMUserCollectionToDistributionPointGroup -DistributionPointGroupName <String> -  
UserCollectionId <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: AddUserCollectionToDistributionPointGroupById_Object

```
Add-CMUserCollectionToDistributionPointGroup -DistributionPointGroup <IResultObject> -  
UserCollectionId <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: AddUserCollectionToDistributionPointGroupByName_Id

```
Add-CMUserCollectionToDistributionPointGroup -DistributionPointGroupId <String> -  
UserCollectionName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: AddUserCollectionToDistributionPointGroupByName_Name

```
Add-CMUserCollectionToDistributionPointGroup -DistributionPointGroupName <String> -  
UserCollectionName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: AddUserCollectionToDistributionPointGroupByName_Object

```
Add-CMUserCollectionToDistributionPointGroup -DistributionPointGroup <IResultObject> -  
UserCollectionName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: AddUserCollectionToDistributionPointGroupByObject_Id

```
Add-CMUserCollectionToDistributionPointGroup -DistributionPointGroupId <String> -  
UserCollection <IResultObject> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: AddUserCollectionToDistributionPointGroupByObject_Name

```
Add-CMUserCollectionToDistributionPointGroup -DistributionPointGroupName <String> -  
UserCollection <IResultObject> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: AddUserCollectionToDistributionPointGroupByObject_Object

```
Add-CMUserCollectionToDistributionPointGroup -DistributionPointGroup <IResultObject> -  
UserCollection <IResultObject> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Add-CMUserCollectionToDistributionPointGroup** cmdlet associates members of a user collection to a distribution point group. A distribution point group is a set of distribution points that you can manage as a single entity.

After you distribute content to a collection, and then associate the collection to a new distribution point group, you must redistribute the content to the collection before the content is distributed to the new distribution point group.

Parameters

-DistributionPointGroup<IRResultObject>

Specifies a **CMDistributionPointGroup** object. To obtain a **CMDistributionPointGroup** object, use the **Get-CMDistributionPointGroup** cmdlet.

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

-DistributionPointGroupId<String>

Specifies an ID of a distribution point group.

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

-DistributionPointGroupName<String>

Specifies the name of a distribution point group.

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

-UserCollection<IResultObject>

Specifies a **CMUserCollection** object. To obtain a **CMUserCollection** object, use the **Get-
CMUserCollection** cmdlet.

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

-UserCollectionId<String>

Specifies an ID of a user collection.

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

-UserCollectionName<String>

Specifies the name of a user collection.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Associate a user collection with a distribution point group by using a name

This command associates the user collection named UserCollection05 with the distribution point group named DPG05ContosoWest.

```
PS C:\> Add-CMDeviceCollectionToDistributionPointGroup -UserCollectionName  
"UserCollection05" -DistributionPointGroupName "DPG05ContosoWest"
```

Related topics

[Get-CMDistributionPointGroup](#)

[Get-CMUserCollection](#)

Approve-CMApprovalRequest

Approve-CMApprovalRequest

Approves a request to allow the installation of an application.

Syntax

Parameter Set: SearchByIdMandatory

```
Approve-CMApprovalRequest -Id <String[]> [-Comment <String> ] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByNameMandatory

```
Approve-CMApprovalRequest -ApplicationName <String[]> -User <String> [-Comment <String> ] [-  
Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Approve-CMApprovalRequest -InputObject <IResultObject> [-Comment <String> ] [-Confirm] [-  
WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Approve-CMApprovalRequest** cmdlet approves a request from a user to install an application. You can specify an approval request by application name, application ID, or by user. You can also use the **Get-CMApprovalRequest** cmdlet to view approval requests.

Parameters

-ApplicationName<String[]>

Specifies an array of names of applications.

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

-Comment<String>

Specifies a comment about the approval of the request.

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

-Id<String[]>

Specifies an array of IDs of applications.

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

-InputObject<IResultObject>

Specifies an approval request object. To obtain an approval request object, use the [Get-CMAApprovalRequest](#) cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

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

-User<String>

Specifies the name of a user who submitted the approval request. Use the format *domain\user*.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Approve a request for a specific application

This command approves a request from a user to install an application specified by its ID.

```
PS C:\> Approve-CMApprovalRequest -Id "ScopeId_2A11048C-917A-4C11-9E77-7DCC402F30EC/Application_426dfca1-0cc0-4aa3-85f8-3cd1b184d494/1"
```

Example 2: Approve a request for a specific user

This command approves a request for an application named Test for the specified user. The command includes a comment.

```
PS C:\> Approve-CMApprovalRequest -Application "Test" -User "tsqa\davidchew" -Comment "Request approved."
```

Example 3: Approve a request by using a variable

The first command gets an approval request for a specified application ID and stores it in the variable \$Approval.

The second command accepts the request stored in \$Approval. The command includes a comment.

```
PS C:\> $Approval = Get-CMApprovalRequest -Id "ScopeId_2A11048C-917A-4C11-9E77-7DCC402F30EC/Application_d047e945-d6af-46f4-910f-ed36c880ae06/1"
PS C:\> Approve-CMApprovalRequest -InputObject $Approval -Comment "Request approved."
```

Related topics

[Deny-CMApprovalRequest](#)

[Get-CMApprovalRequest](#)

Approve-CMDevice

Approve-CMDevice

Approves Configuration Manager device clients.

Syntax

Parameter Set: SearchByNameMandatory

```
Approve-CMDevice -DeviceName <String[]> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Approve-CMDevice -DeviceId <String[]> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Approve-CMDevice -InputObject <IResultObject> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Approve-CMDevice** cmdlet approves one or more Microsoft System Center 2012 Configuration Manager device clients to join a site. You cannot approve a System Center 2012 Configuration Manager client until you have installed the device and assigned it to a site.

When the client device communicates with site systems by using HTTP and a self-signed certificate, you must approve these clients to identify them as trusted computers. By default, the site configuration automatically approves client devices from the same Active Directory forest and trusted forests, so you do not have to manually approve each client device. You must manually approve workgroup computers that you trust and any other computers that you trust but are not approved.

You do not have to approve clients that always communicate to site systems by using HTTPS rather than HTTP, or clients that use a PKI certificate when they communicate to site systems by using HTTP. These clients establish trust with System Center 2012 Configuration Manager by using the public key infrastructure (PKI) certificates.

Parameters

-DeviceId<String[]>

Specifies an array of device IDs.

Aliases	ResourceId
---------	------------

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

-DeviceName<String[]>

Specifies an array of device names.

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

-InputObject<IResultObject>

Specifies a **CMDevice** object. To obtain a **CMDevice** object, use the [Get-CMDevice](#) cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Approve a device client

This command approves the Configuration Manager device client named CMCEN-DIST02.

```
PS C:\> Approve-CMDevice -DeviceName "CMCEN-DIST02"
```

Related topics

[Block-CMDevice](#)

[Remove-CMDevice](#)

[Get-CMDevice](#)

[Unblock-CMDevice](#)

Approve-CMUserDeviceAffinityRequest

Approve-CMUserDeviceAffinityRequest

Approves a request for user device affinity in Configuration Manager.

Syntax

Parameter Set: SearchByNameMandatory

```
Approve-CMUserDeviceAffinityRequest -CollectionName <String> [-DeviceId <String> ] [-DeviceName <String> ] [-UserDeviceAffinityRequest <IResultObject> ] [-UserDeviceAffinityRequestId <String> ] [-UserId <String> ] [-UserName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Approve-CMUserDeviceAffinityRequest -CollectionId <String> [-DeviceId <String> ] [-DeviceName <String> ] [-UserDeviceAffinityRequest <IResultObject> ] [-UserDeviceAffinityRequestId <String> ] [-UserId <String> ] [-UserName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Approve-CMUserDeviceAffinityRequest** cmdlet approves a request for user device affinity.

In Microsoft System Center 2012 Configuration Manager, user device affinity defines a relationship between a user and a device. Instead of deploying an application to a group of devices, you deploy an application to a user and Configuration Manager installs the application on all devices associated with the user.

Parameters

-CollectionId<String>

Specifies a collection ID that represents the user device affinity in Configuration Manager.

Aliases	none
Required?	true
Position?	named
Default Value	none

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

-CollectionName<String>

Specifies a name of a collection that represents the user device affinity in Configuration Manager.

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

-DeviceId<String>

Specifies a device ID in Configuration Manager.

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

-DeviceName<String>

Specifies a device name in Configuration Manager.

Aliases	none
Required?	false
Position?	named
Default Value	none

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

-UserDeviceAffinityRequest<IResultObject>

Specifies a **CMUserDeviceAffinityRequest** object. To obtain a **CMUserDeviceAffinityRequest** object, use the **Get-CMUserDeviceAffinityRequest** cmdlet.

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

-UserDeviceAffinityRequestId<String>

Specifies a unique ID for a request for user device affinity.

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

-UserId<String>

Specifies a Configuration Manager ID for a user resource.

Aliases	none
Required?	false
Position?	named

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

-UserName<String>

Specifies a user name for a resource in Configuration Manager.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Approve a request for user device affinity

This command approves a request for user device affinity for the collection named All Systems.

```
PS C:\> Approve-CMUserDeviceAffinityRequest -CollectionName "All Systems" -UserName "Western\EvanNarvaez$"
```

Related topics

[Deny-CMUserDeviceAffinityRequest](#)

[Get-CMUserDeviceAffinityRequest](#)

Block-CMCertificate

Block-CMCertificate

Blocks a certificate.

Syntax

Parameter Set: SearchByIdMandatory

```
Block-CMCertificate -Id <String[]> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Block-CMCertificate** cmdlet blocks a certificate. Microsoft System Center 2012 Configuration Manager uses certificates to manage boot media, Pre-Boot EXecution Environment (PXE) deployments, and Independent Software Vendors (ISV) proxies.

Parameters

-Id<String[]>

Specifies an array of certificate IDs.

Aliases	Smsid
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Block a certificate

This command blocks the certificate that has the specified ID.

```
PS C:\> Block-CMCertificate -Id "11729"
```

Related topics

[Import-CMCertificate](#)

[Unblock-CMCertificate](#)

[Update-CMCertificate](#)

Block-CMConflictingRecord

Block-CMConflictingRecord

Creates a blocked Configuration Manager record for client that has a conflicting record.

Syntax

Parameter Set: SearchByNameMandatory

```
Block-CMConflictingRecord -Name <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Block-CMConflictingRecord -Id <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Block-CMConflictingRecord -ConflictingRecord <IResultObject> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Block-CMConflictingRecord** cmdlet blocks a record for a client that has a conflicting record in Microsoft System Center 2012 Configuration Manager.

When System Center 2012 Configuration Manager recognizes a new client, it uses hardware information to check whether it previously created a record for that computer. For example, you might have reinstalled the operating system. The previous client record still exists with the same hardware information. If you manually resolve conflicts, you have the option to merge the new record with the existing record, create a new record, or create a record as a blocked record. You can also configure System Center 2012 Configuration Manager to resolve conflicts automatically.

You can specify a conflict by using a name or ID or you can use the [Get-CMConflictingRecord](#) cmdlet to obtain one.

Parameters

-ConflictingRecord<IResultObject>

Specifies a conflicting record object. To obtain a conflicting record object, use the [Get-CMConflictingRecord](#) cmdlet.

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

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

-Id<String>

Specifies an ID for the conflicting records.

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

-Name<String>

Specifies a name for the conflicting records.

Aliases	AgentName
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Create a blocked record for a named conflict

This command creates a blocked record for the conflict named CR07.

```
PS C:\> Block-CMConflictingRecord -Name "CR07"
```

Example 2: Create a blocked record by using a variable

The first command gets a conflicting record named CR07 and saves it in the \$CMCR variable.

The second command creates a blocked record for the conflict in \$CRCM.

```
PS C:\> $CMCR = Get-CMConflictingRecord -Name "CR07"
PS C:\> Block-CMConflictingRecord -ConflictingRecord $CMCR
```

Related topics

[Get-CMConflictingRecord](#)

[Merge-CMConflictingRecord](#)

Block-CMDevice

Block-CMDevice

Blocks Configuration Manager client devices.

Syntax

Parameter Set: SearchByNameMandatory

```
Block-CMDevice -DeviceName <String[]> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Block-CMDevice -DeviceId <String[]> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Block-CMDevice -InputObject <IResultObject> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Block-CMDevice** cmdlet blocks one or more client devices. You must block a device from the client's assigned site. You cannot block the device from sites higher in the hierarchy. Blocked devices are ignored by the Microsoft System Center 2012 Configuration Manager hierarchy.

Block a device that you no longer trust, to prevent it from receiving client policy and to prevent System Center 2012 Configuration Manager site systems from communicating with it.

If you later change your mind, you can unblock a device that has been blocked. If you unblock an Intel Active Management Technology (AMT)-based computer that you provisioned for AMT when it was blocked, you must take additional steps before you can manage that computer again out-of-band.

For more information about unblocking a client device in System Center 2012 Configuration Manager, see [Determine Whether to Block Clients in Configuration Manager](http://go.microsoft.com/fwlink/?LinkId=271116) (http://go.microsoft.com/fwlink/?LinkId=271116) on TechNet.

Parameters

-DeviceId<String[]>

Specifies an array of device IDs.

Aliases	ResourceId
Required?	true

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeviceName<String[]>

Specifies an array of device names.

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

-InputObject<IResultObject>

Specifies a **CMDevice** object. To obtain a **CMDevice** object, use the **Get-CMDevice** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Block a device client

This command blocks the Configuration Manager device client named CMCEN-DIST02.

```
PS C:\> Block-CMDevice -DeviceName "CMCEN-DIST02"
```

Related topics

[Unblock-CMDevice](#)

[Get-CMDevice](#)

[Remove-CMDevice](#)

[Approve-CMDevice](#)

Clear-CMAmtAuditLog

Clear-CMAmtAuditLog

Clears audit log entries for Intel AMT-based computers.

Syntax

Parameter Set: SearchByNameMandatory

```
Clear-CMAmtAuditLog -DeviceName <String[]> [-Force] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByCollectionIdMandatory

```
Clear-CMAmtAuditLog -DeviceCollectionId <String[]> [-Force] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByCollectionNameMandatory

```
Clear-CMAmtAuditLog -DeviceCollectionName <String[]> [-Force] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByCollectionValueMandatory

```
Clear-CMAmtAuditLog -DeviceCollection <IResultObject> [-Force] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Clear-CMAmtAuditLog -DeviceId <String[]> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Clear-CMAmtAuditLog -Device <IResultObject> [-Force] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Detailed Description

The **Clear-CMAmtAuditLog** cmdlet clears audit log entries for Intel Active Management Technology (Intel AMT)-based computers. The audit log records authorized and authenticated out-of-band management activities performed on Intel AMT computers.

Depending on your Intel AMT version, once the audit log becomes 85 percent full, noncritical log entries might not be written to the log or might overwrite old entries. This cmdlet does not stop audit logging.

You can use the **Disable-CMAmtAuditLog** cmdlet to stop logging.

You can specify computers by using the Microsoft System Center 2012 Configuration Manager device name or device ID, or you can use the **Get-CMDevice** cmdlet to get a device object. You can also clear audit logs for all the devices in a System Center 2012 Configuration Manager collection. Specify a

collection by using the collection name or collection ID, or you can use the **Get-CMDeviceCollection** cmdlet to get a device collection object.

Parameters

-Device<IResultObject>

Specifies a device object. To obtain a device object, use the **Get-CMDevice** cmdlet.

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

-DeviceCollection<IResultObject>

Specifies a device collection object. To obtain a device collection object, use the **Get-CMDeviceCollection** cmdlet.

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

-DeviceCollectionId<String[]>

Specifies an array of IDs of device collections.

Aliases	none
Required?	true
Position?	named

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

-DeviceCollectionName<String[]>

Specifies an array of names of device collections.

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

-DeviceId<String[]>

Specifies an array of IDs of devices.

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

-DeviceName<String[]>

Specifies an array of names of devices.

Aliases	Name
Required?	true
Position?	named

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

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Clear the audit log by using an ID

This command clears the Intel AMT audit log for a device that has the ID 16777230.

```
PS C:\> Clear-CMAmtAuditLog -DeviceID "16777230"
```

Example 2: Clear audit logs for a device collection

This command clears Intel AMT audit logs for the devices in a collection named Floor03.

```
PS C:\> Clear-CMAmtAuditLog -DeviceCollectionName "Floor03"
```

Example 3: Clear the audit log by using a variable

The first command gets a device object by using the **Get-CMDevice** cmdlet, and stores it in the \$CMD variable.

The second command clears the Intel AMT audit for the device stored in the \$CMD variable. The command uses the *Force* parameter. Therefore, the command does not prompt you for confirmation.

```
PS C:\> $CMD = Get-CMDevice -Name "Accn023.Contoso.com"  
PS C:\> Clear-CMAmtAuditLog -Device $CMD -Force
```

Related topics

[Disable-CMAmtAuditLog](#)

[Enable-CMAmtAuditLog](#)

[Get-CMDevice](#)

[Get-CMDeviceCollection](#)

Clear-CMClientOperation

Clear-CMClientOperation

Clears a Configuration Manager client operation object.

Syntax

Parameter Set: ClearById

```
Clear-CMClientOperation -Id <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Clear-CMClientOperation** cmdlet clears a Microsoft System Center 2012 Configuration Manager client operation object. Specify the operation to clear by using its ID.

You can use the **Remove-CMClientOperation** cmdlet to remove a client operation object.

Parameters

-Id<String>

Specifies the ID of a client operation.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Clear a client operation

This command clears the client operation that has the ID CMCO217.

```
PS C:\> Clear-CMClientOperation -Id "CMCO217"
```

Related topics

[Remove-CMClientOperation](#)

[Invoke-CMClientOperationSummarization](#)

Clear-CMComponentStatusMessageCount

Clear-CMComponentStatusMessageCount

Changes the component status message count to zero.

Syntax

Parameter Set: SearchByName

```
Clear-CMComponentStatusMessageCount -ComponentName <String> -Severity {All | Error | Information | Warning} -SiteCode <String> [-ComputerName <String> ] [-Confirm] [-WhatIf] [<CommonParameters>]
```

Detailed Description

The **Clear-CMComponentStatusMessageCount** cmdlet changes the component status message count to zero (0).

Microsoft System Center 2012 Configuration Manager indicates whether operations succeed or fail and include other information in component status messages. Threads or processes send component status messages to System Center 2012 Configuration Manager sites, identified by site codes.

You can define which message count to set to zero by the component that created the messages, severity of the messages, and the site code of the System Center 2012 Configuration Manager server that receives the messages. You can also specify the computer that hosts that component.

Parameters

-ComponentName<String>

Specifies the name of a component that creates messages.

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

-ComputerName<String>

Specifies the name of a computer that hosts the component.

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

-Severity<Severity>

Specifies the severity of a component status message. Valid values are:

- All
- Error
- Information
- Warning

The acceptable values for this parameter are:

All	
Error	
Information	
Warning	

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

-SiteCode<String>

Specifies the site code for a Configuration Manager site. Status messages originate in this site.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Clear message count

This command changes the message count to zero for the component SMS_HIERARCHY_MANAGER for all message severity types. The command specifies the site that has the site code CM1.

```
PS C:\> Clear-CMComponentStatusMessageCount -ComponentName "SMS_HIERARCHY_MANAGER" -Severity ALL -SiteCode "CM1"
```

Example 2: Clear error message count

This command changes the message count to zero for the component SMS_DISTRIBUTION_MANAGER for error messages. The command specifies the site that has the site code CM1, and includes the computer name West34.Western.Contoso.com.

```
PS C:\> Clear-CMComponentStatusMessageCount -ComponentName "SMS_DISTRIBUTION_MANAGER" -Severity Error -SiteCode "CM1" -ComputerName "West34.Western.Contoso.com"
```

Related topics

[Get-CMComponentStatusMessage](#)

Clear-CMMigrationData

Clear-CMMigrationData

Deletes historical data about a data migration operation.

Syntax

Parameter Set: CleanupMigrationDataBySiteCode

```
Clear-CMMigrationData -SiteCode <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Clear-CMMigrationData** cmdlet deletes the historical data about a data migration operation. With Microsoft System Center 2012 Configuration Manager, you can migrate data from a supported Configuration Manager hierarchy to a System Center 2012 Configuration Manager environment. When you migrate data from a source hierarchy, you access data from the site databases that you identify in the source infrastructure and then transfer that data to your current environment from the database of the destination hierarchy. **Clear-CMMigrationData** cleans up the historical data from the destination hierarchy database.

Parameters

-Force

Performs the action without a confirmation message.

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

-SiteCode<String>

Specifies the site code of the destination site from which you clear historical data.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Clean up historical data from a migration

This command removes the historical data from the destination site that has the site code C04.

```
PS C:\> Clear-CMMigrationData -SiteCode "C04"
```

Related topics

[Set-CMMigrationExclusionList](#)

[Set-CMMigrationSource](#)

Clear-CMOperatingSystemImageUpdateSchedule

Clear-CMOperatingSystemImageUpdateSchedule

Removes a schedule for updating an operating system image.

Syntax

Parameter Set: SearchByNameMandatory

```
Clear-CMOperatingSystemImageUpdateSchedule -OperatingSystemImageName <String[]> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Clear-CMOperatingSystemImageUpdateSchedule -OperatingSystemImageId <String[]> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Clear-CMOperatingSystemImageUpdateSchedule -OperatingSystemImage <IResultObject> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Clear-CMOperatingSystemImageUpdateSchedule** cmdlet removes a schedule for updating an operating system image from a Microsoft System Center 2012 Configuration Manager site.

Operating system images are .wim format files, which represent a compressed collection of reference files and folders that System Center 2012 Configuration Manager requires to successfully install and configure an operating system on a computer. You can use System Center 2012 Configuration Manager to define a schedule for updating these images by using Component Based Servicing (CBS), then delete unwanted schedules by using this cmdlet.

Parameters

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OperatingSystemImage<IResultObject>

Specifies the update schedule to remove by using an update schedule object. To obtain an object for an update schedule, use the **Get-CMOperatingSystemImageUpdateSchedule** cmdlet.

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

-OperatingSystemImageId<String[]>

Specifies an array of IDs of operating system images.

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

-OperatingSystemImageName<String[]>

Specifies an array of names of operating system images.

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

Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Remove a schedule for updating an operating system image by using a name

This command removes a schedule named Win8UpdateSchedule that updates an operating system image.

```
PS C:\> Clear-CMOperatingSystemUpdateSchedule -OperatingSystemImageName "Win8UpdateSchedule"
```

Example 2: Remove a schedule for updating an operating system image by using an object

The first command gets the image update schedule by using the ID 1207 and puts this schedule in the variable named \$UpdateSchedObject.

The second command removes the image update schedule by using the object in the \$UpdateSchedObject variable.

```
PS C:\> $Win8UpdateSchedule = Get-CMOperatingSystemUpdateSchedule -Id 1207
PS C:\> Clear-CMOperatingSystemImageUpdateSchedule -OperatingSystemImageName
"Win8UpdateSchedule"
```

Related topics

[Get-CMOperatingSystemImageUpdateSchedule](#)

[Set-CMOperatingSystemImageUpdateSchedule](#)

Clear-CMPxeDeployment

Clear-CMPxeDeployment

Clears the status of the most recent PXE deployment in Configuration Manager.

Syntax

Parameter Set: SearchByNameMandatory

```
Clear-CMPxeDeployment -DeviceCollectionName <String[]> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Clear-CMPxeDeployment -DeviceCollectionId <String[]> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByIdMandatory__Device

```
Clear-CMPxeDeployment -ResourceId <String[]> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByNameMandatory_Device

```
Clear-CMPxeDeployment -DeviceName <String[]> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Clear-CMPxeDeployment -DeviceCollection <IResultObject> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByValueMandatory__Device

```
Clear-CMPxeDeployment -Device <IResultObject> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Clear-CMPxeDeployment** cmdlet clears the status of the most recent Preboot Execution Environment (PXE) deployment in Microsoft System Center 2012 Configuration Manager.

You can redeploy a required PXE deployment for a collection of devices. Clear the status of the last PXE deployment assigned to that System Center 2012 Configuration Manager collection. System Center 2012 Configuration Manager redeploys the most recent required deployments.

Parameters

-Device<IResultObject>

Specifies a device object. To obtain a device object, use the **Get-CMDevice** cmdlet.

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

-DeviceCollection<IResultObject>

Specifies a device collection object. To obtain a device collection object, use the **Get-CMDeviceCollection** cmdlet.

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

-DeviceCollectionId<String[]>

Specifies an array of IDs of device collections.

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

-DeviceCollectionName<String[]>

Specifies an array of names of device collections.

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

-DeviceName<String[]>

Specifies an array of names of devices.

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

-ResourceId<String[]>

Specifies an array of IDs for resources. The cmdlet clears the status of the PXE deployment for these resources.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Clear a PXE deployment for a device collection

This command clears a PXE deployment identified with a device collection ID.

```
PS C:\> Clear-CMPxeDeployment -DeviceCollectionId "SMS00072"
```

Related topics

[Get-CMDeviceCollection](#)

[Get-CMDevice](#)

Clear-CMSiteStatusMessageCount

Clear-CMSiteStatusMessageCount

Clears the message count in Configuration Manager.

Syntax

Parameter Set: SearchByName

```
Clear-CMSiteStatusMessageCount -ComputerName <String> -Severity {All | Error | Information | Warning} -SiteCode <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Clear-CMSiteStatusMessageCount** cmdlet clears the message count in Microsoft System Center 2012 Configuration Manager.

Parameters

-ComputerName<String>

Specifies the name of a computer in Configuration Manager.

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

-Severity<Severity>

Specifies a message severity. Valid values are: All, Error, Information, and Warning.

The acceptable values for this parameter are:

All	
Error	
Information	
Warning	

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

-SiteCode<String>

Specifies a site code in Configuration Manager.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Clear the status message count

This command clears the error message count for the computer.

```
PS C:\> Clear-CMSiteStatusMessageCount -ComputerName "Contoso-Test" -Severity Error -  
SiteCode "CM1"
```

Related topics

[Get-CMSiteStatusMessage](#)

Convert-CMSchedule

Convert-CMSchedule

Converts schedule tokens into and from interval strings.

Syntax

Parameter Set: ByToken

```
Convert-CMSchedule [-ScheduleToken] <IResultObject[]> [ <CommonParameters>]
```

Parameter Set: ByString

```
Convert-CMSchedule [-ScheduleString] <String[]> [ <CommonParameters>]
```

Detailed Description

The **Convert-CMSchedule** cmdlet decodes and encodes schedule tokens into and from a Microsoft System Center 2012 Configuration Manager interval strings.

Parameters

-ScheduleString<String[]>

Specifies an array of interval strings.

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

-ScheduleToken<IResultObject[]>

Specifies an array of Configuration Manager schedule objects output from another cmdlet.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

[New-CMSchedule](#)

[Get-CMClientStatusUpdateSchedule](#)

[Get-CMBaselineSummarizationSchedule](#)

[Get-CMOperatingSystemImageUpdateSchedule](#)

[Get-CMEndpointProtectionSummarizationSchedule](#)

[Get-CMSoftwareUpdateSummarizationSchedule](#)

Copy- CMClientAuthCertificateProfileConfigurationItem

Copy-CMClientAuthCertificateProfileConfigurationItem

Renames a certificate profile.

Syntax

Parameter Set: SearchByNameMandatory

```
Copy-CMClientAuthCertificateProfileConfigurationItem -Name <String[]> -NewName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Copy-CMClientAuthCertificateProfileConfigurationItem -Id <String[]> -NewName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Copy-CMClientAuthCertificateProfileConfigurationItem -InputObject <IResultObject> -NewName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Copy-CMClientAuthCertificateProfileConfigurationItem** cmdlet renames a certificate profile. Client computers use certificate profiles to authenticate when they use services such as a virtual private network (VPN) or a wireless network.

Parameters

-Id<String[]>

Specifies an array of IDs of certificate profiles.

Aliases	CId
Required?	true
Position?	named

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

-InputObject<IResultObject>

Specifies a certificate profile object. To obtain a certificate profile object, use the **Get-
CMClientAuthCertificateProfileConfigurationItem** cmdlet.

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

-Name<String[]>

Specifies an array of names of certificate profiles.

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

-NewName<String>

Specifies a new name for the certificate profile.

Aliases	none
Required?	true

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Rename a certificate profile

This command renames the certificate profile named VPN_Certificate.

```
PS C:\> Copy-CMClientAuthCertificateProfileConfigurationItem -Name "VPN_Certificate" -
NewName "VPN_Certificate_Contoso"
```

Related topics

[Get-CMClientAuthCertificateProfileConfigurationItem](#)

[New-CMClientAuthCertificateProfileConfigurationItem](#)

[Remove-CMClientAuthCertificateProfileConfigurationItem](#)

[Set-CMClientAuthCertificateProfileConfigurationItem](#)

Copy-CMRemoteConnectionProfileConfigurationItem

Copy-CMRemoteConnectionProfileConfigurationItem

Renames a remote connection profile.

Syntax

Parameter Set: SearchByNameMandatory

```
Copy-CMRemoteConnectionProfileConfigurationItem -Name <String[]> -NewName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Copy-CMRemoteConnectionProfileConfigurationItem -Id <String[]> -NewName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Copy-CMRemoteConnectionProfileConfigurationItem -InputObject <IResultObject> -NewName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Copy-CMRemoteConnectionProfileConfigurationItem** cmdlet renames a remote connection profile.

Parameters

-Id<String[]>

Specifies an array of IDs for remote connection profiles.

Aliases	CId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

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

-InputObject<IResultObject>

Specifies a remote connection profile object. To obtain a remote connection profile, use the **Get-CMRemoteConnectionProfileConfigurationItem** cmdlet.

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

-Name<String[]>

Specifies an array of names of remote connection profiles.

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

-NewName<String>

Specifies a new name for the remote connection profile.

Aliases	none
Required?	true
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Rename a profile

This command renames the remote connection profile named Corp_RD_Gateway.

```
PS C:\> Copy-CMRemoteConnectionProfileConfigurationItem -Name "Corp_RD_Gateway" -NewName "Corp_Remote_Gateway"
```

Related topics

[Get-CMRemoteConnectionProfileConfigurationItem](#)

[New-CMRemoteConnectionProfileConfigurationItem](#)

[Remove-CMRemoteConnectionProfileConfigurationItem](#)

[Set-CMRemoteConnectionProfileConfigurationItem](#)

Copy-CMSecurityRole

Copy-CMSecurityRole

Creates a custom security role.

Syntax

Parameter Set: CopyFromId

```
Copy-CMSecurityRole -Name <String> -SourceRoleId <String> [-Description <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: CopyFromName

```
Copy-CMSecurityRole -Name <String> -SourceRoleName <String> [-Description <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: CopyFromValue

```
Copy-CMSecurityRole -InputObject <IResultObject> -Name <String> [-Description <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Copy-CMSecurityRole** cmdlet creates a new security role by using an existing security role as a template. Microsoft System Center 2012 Configuration Manager provides several built-in security roles. If you require additional security roles, you can create a custom security role by creating a copy of an existing security role, and then modifying the copy.

Parameters

-Description<String>

Specifies the description of a security role.

Aliases	RoleDescription
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

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

-InputObject<IResultObject>

Specifies a **CMSecurityRole** object. To obtain a **CMSecurityRole** object, use the [Get-CMSecurityRole](#) cmdlet.

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

-Name<String>

Specifies a name for the new security scope.

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

-SourceRoleId<String>

Specifies the ID of a security role.

Aliases	CopiedFromId
Required?	true
Position?	named
Default Value	none

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

-SourceRoleName<String>

Specifies the name of a security role.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Copy a security role by using an ID

This command creates a new security role named SecRole02 by copying the security role that has the ID SMS000CR.

```
PS C:\> Copy-CMSecurityRole -Name "SecRole02" -SourceRoleId "SMS000CR"
```

Example 2: Copy a security role by using a name

This command creates a new security role named SecRole02 by copying the security role named Software Update Manager.

```
PS C:\> Copy-CMSecurityRole -Name "SecRole02" -SourceRoleName "Software Update Manager"
```

Example 3: Copy a security role

The first command gets the security role named Software Update Manager and stores it in the \$Srole variable.

The second command creates a new security role named SecRole02 by copying the object stored in \$Srole.

```
PS C:\> $Srole = Get-CMSecurityRole -Name "Software Update Manager"  
PS C:\> Copy-CMSecurityRole -InputObject $Srole -Name "SecRole02"
```

Related topics

[Get-CMSecurityRole](#)

[Set-CMSecurityRole](#)

[Remove-CMSecurityRole](#)

[Import-CMSecurityRole](#)

[Export-CMSecurityRole](#)

[Remove-CMSecurityRoleFromAdministrativeUser](#)

Copy- CMTrustedRootCertificateProfileConfigurationItem

Copy-CMTrustedRootCertificateProfileConfigurationItem

Renames a root certificate profile.

Syntax

Parameter Set: SearchByNameMandatory

```
Copy-CMTrustedRootCertificateProfileConfigurationItem -Name <String[]> -NewName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Copy-CMTrustedRootCertificateProfileConfigurationItem -Id <String[]> -NewName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Copy-CMTrustedRootCertificateProfileConfigurationItem -InputObject <IResultObject> -NewName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Copy-CMTrustedRootCertificateProfileConfigurationItem** cmdlet renames a root certificate profile.

Parameters

-Id<String[]>

Specifies an array of IDs of root certificate profiles.

Aliases	CId
Required?	true
Position?	named
Default Value	none

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

-InputObject<IResultObject>

Specifies a root certificate profile object. To obtain a root certificate profile object use the **Get-
CMTrustedRootCertificateProfileConfigurationItem** cmdlet.

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

-Name<String[]>

Specifies an array of names of root certificate profiles.

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

-NewName<String>

Specifies the new name for the root certificate profile.

Aliases	none
Required?	true
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Rename a root certificate profile

This command renames the root certificate profile named Corp_Root_Cert.

```
PS C:\> Copy-CMRootCertificateProfileConfigurationItem -Name "Corp_Root_Cert" -NewName  
"Corp_Production_RootCert"
```

Related topics

[Get-CMTrustedRootCertificateProfileConfigurationItem](#)

[New-CMTrustedRootCertificateProfileConfigurationItem](#)

[Remove-CMTrustedRootCertificateProfileConfigurationItem](#)

[Set-CMTrustedRootCertificateProfileConfigurationItem](#)

Copy-CMUserDataAndProfileConfigurationItem

Copy-CMUserDataAndProfileConfigurationItem

Creates a copy of a user data and profile configuration item.

Syntax

Parameter Set: SearchByNameMandatory

```
Copy-CMUserDataAndProfileConfigurationItem -Name <String[]> -NewName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Copy-CMUserDataAndProfileConfigurationItem -Id <String[]> -NewName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Copy-CMUserDataAndProfileConfigurationItem -InputObject <IResultObject> -NewName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Copy-CMUserDataAndProfileConfigurationItem** cmdlet creates a copy of a user data and profile configuration item. The new configuration item does not retain any relationship to the original configuration item. This means that the duplicate configuration item does not continue to inherit configuration information from the original configuration item.

User data and profile configuration items in Microsoft System Center 2012 Configuration Manager contain settings that can manage folder redirection, offline files, and roaming profiles on computers that run Windows® 8 for users in your hierarchy.

Parameters

-Id<String[]>

Specifies an array of IDs of user data and profile configuration items.

Aliases	CId
Required?	true
Position?	named

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

-InputObject<IResultObject>

Specifies a **CMUserDataAndProfileConfigurationItem** object. To obtain a **CMUserDataAndProfileConfigurationItem** object, use the **Get-
CMUserDataAndProfileConfigurationItem** cmdlet.

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

-Name<String[]>

Specifies an array of names of user data and profile configuration items.

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

-NewName<String>

Specifies the name of the new user data and profile configuration item.

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

Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Copy a user data and profile configuration item

This command creates a copy of the user data and profile configuration item named D01ConfigItem. The new user data and profile configuration item is named UDPConfig02.

```
PS C:\> Copy-CMUserDataAndProfileConfigurationItem -NewName "UDPConfig02" -Name "D01ConfigItem"
```

Related topics

[Get-CMUserDataAndProfileConfigurationItem](#)

[Remove-CMUserDataAndProfileConfigurationItem](#)

[Set-CMUserDataAndProfileConfigurationItem](#)

Copy-CMVpnProfileConfigurationItem

Copy-CMVpnProfileConfigurationItem

Renames a VPN profile.

Syntax

Parameter Set: SearchByNameMandatory

```
Copy-CMVpnProfileConfigurationItem -Name <String[]> -NewName <String> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Copy-CMVpnProfileConfigurationItem -Id <String[]> -NewName <String> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Copy-CMVpnProfileConfigurationItem -InputObject <IResultObject> -NewName <String> [-Confirm] [  
-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Copy-CMVpnProfileConfigurationItem** cmdlet renames a virtual private network (VPN) profile.

Parameters

-Id<String[]>

Specifies an array of IDs of VPN profile objects.

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

-InputObject<IResultObject>

Specifies a VPN profile object. To obtain a VPN profile object, use the **Get-CMVpnProfileConfigurationItem** cmdlet.

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

-Name<String[]>

Specifies an array of names of VPN profiles.

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

-NewName<String>

Specifies the new name to assign to the VPN profile.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Rename a VPN profile

This command renames the VPN profile named Corp_VPNNet1.

```
PS C:\> Copy-CMVpnProfileConfigurationItem -Name "Corp_VPNNet1" -NewName  
"Corp_Production_VPN"
```

Related topics

[Get-CMVpnProfileConfigurationItem](#)

[New-CMVpnProfileConfigurationItem](#)

[Remove-CMVpnProfileConfigurationItem](#)

[Set-CMVpnProfileConfigurationItem](#)

Copy-CMWirelessProfileConfigurationItem

Copy-CMWirelessProfileConfigurationItem

Renames a wireless profile.

Syntax

Parameter Set: SearchByNameMandatory

```
Copy-CMWirelessProfileConfigurationItem -Name <String[]> -NewName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Copy-CMWirelessProfileConfigurationItem -Id <String[]> -NewName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Copy-CMWirelessProfileConfigurationItem -InputObject <IResultObject> -NewName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Copy-CMWirelessProfileConfigurationItem** cmdlet renames a wireless profile.

Parameters

-Id<String[]>

Specifies an array of IDs of wireless profiles.

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

-InputObject<IResultObject>

Specifies a wireless profile object. To obtain a wireless profile object, use the **Get-CMWirelessProfileConfigurationItem** cmdlet.

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

-Name<String[]>

Specifies an array of names of wireless profiles.

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

-NewName<String>

Specifies the new name for the wireless profile.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Rename a wireless profile

This command renames the wireless profile named AP_Test.

```
PS C:\> Copy-CMWirelessProfileConfigurationItem -Name "AP_Test" -NewName "AP_Production"
```

Related topics

[Get-CMWirelessProfileConfigurationItem](#)

[New-CMWirelessProfileConfigurationItem](#)

[Remove-CMWirelessProfileConfigurationItem](#)

[Set-CMWirelessProfileConfigurationItem](#)

Deny-CMApprovalRequest

Deny-CMApprovalRequest

Denies a request to allow the installation of an application.

Syntax

Parameter Set: SearchByIdMandatory

```
Deny-CMApprovalRequest -Id <String[]> [-Comment <String> ] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByNameMandatory

```
Deny-CMApprovalRequest -ApplicationName <String[]> -User <String> [-Comment <String> ] [-  
Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Deny-CMApprovalRequest -InputObject <IResultObject> [-Comment <String> ] [-Confirm] [-  
WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Deny-CMApprovalRequest** cmdlet denies a request from a user to install an application. You can specify an approval request by application name, application ID, or by user. You can use the **Get-CMApprovalRequest** cmdlet to view approval requests.

Parameters

-ApplicationName<String[]>

Specifies an array of names of applications.

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

-Comment<String>

Specifies a comment about the denial of the request.

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

-Id<String[]>

Specifies an array of IDs of applications.

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

-InputObject<IResultObject>

Specifies an approval request object. To obtain an approval request object, use the [Get-CMAApprovalRequest](#) cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

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

-User<String>

Specifies an array of user names of persons who submitted the approval request. Use the format *domain\user*.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Deny a request by application ID

This command denies a request for an application that has the specified ID. The command includes a comment.

```
PS C:\> Deny-CMApprovalRequest -Id "ScopeId_2A11048C-917A-4C11-9E77-7DCC402F30EC/Application_426dfca1-0cc0-4aa3-85f8-3cd1b184d494/1" -Comment "All requests for this application are denied."
```

Example 2: Deny a request from a specific user

This command denies a request for an application named Test for the specified user.

```
PS C:\> Deny-CMApprovalRequest -Application "Test" -User "tsqa\davidchew"
```

Example 3: Deny a request by using a variable

The first command gets an approval request for a specified application ID and stores it in the \$Approval variable.

The second command denies the request stored in \$Approval. The command includes a comment.

```
PS C:\> $Approval = Get-CMApprovalRequest -Id "ScopeId_2A11048C-917A-4C11-9E77-7DCC402F30EC/Application_426dfca1-0cc0-4aa3-85f8-3cd1b184d494/1"
PS C:\> Deny-CMApprovalRequest -InputObject $Approval -Comment "Request denied."
```

Related topics

[Approve-CMApprovalRequest](#)

[Get-CMApprovalRequest](#)

Deny-CMUserDeviceAffinityRequest

Deny-CMUserDeviceAffinityRequest

Denies a request for user device affinity in Configuration Manager.

Syntax

Parameter Set: SearchByNameMandatory

```
Deny-CMUserDeviceAffinityRequest -CollectionName <String> [-DeviceId <String> ] [-DeviceName <String> ] [-UserDeviceAffinityRequest <IResultObject> ] [-UserDeviceAffinityRequestId <String> ] [-UserId <String> ] [-UserName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Deny-CMUserDeviceAffinityRequest -CollectionId <String> [-DeviceId <String> ] [-DeviceName <String> ] [-UserDeviceAffinityRequest <IResultObject> ] [-UserDeviceAffinityRequestId <String> ] [-UserId <String> ] [-UserName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Deny-CMUserDeviceAffinityRequest** cmdlet denies a request for user device affinity.

In Microsoft System Center 2012 Configuration Manager, user device affinity defines a relationship between a user and a device.

Parameters

-CollectionId<String>

Specifies a collection ID that represents the user device affinity in Configuration Manager.

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

-CollectionName<String>

Specifies a name of a collection that represents the user device affinity in Configuration Manager.

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

-DeviceId<String>

Specifies a device ID in Configuration Manager.

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

-DeviceName<String>

Specifies a device name in Configuration Manager.

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

-UserDeviceAffinityRequest<IResultObject>

Specifies a **CMUserDeviceAffinityRequest** object. To obtain a **CMUserDeviceAffinityRequest** object, use the **Get-CMUserDeviceAffinityRequest** cmdlet.

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

-UserDeviceAffinityRequestId<String>

Specifies a unique ID for a request for user device affinity.

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

-UserId<String>

Specifies a Configuration Manager ID for a user resource.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

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

-UserName<String>

Specifies a user name for a resource in Configuration Manager.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Deny a request for user device affinity

This command denies a request for user device affinity for the collection named All Systems.

```
PS C:\> Deny-CMUserDeviceAffinityRequest -CollectionName "All Systems" -UserName  
"Western\EvanNarvaez$"
```

Related topics

[Approve-CMUserDeviceAffinityRequest](#)

[Get-CMUserDeviceAffinityRequest](#)

Disable-CMAAlert

Disable-CMAAlert

Disables alerts in Configuration Manager.

Syntax

Parameter Set: SearchByIdMandatory

```
Disable-CMAAlert -Id <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByNameMandatory

```
Disable-CMAAlert -Name <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Disable-CMAAlert -InputObject <IResultObject> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Disable-CMAAlert** cmdlet disables one or more alerts in Microsoft System Center 2012 Configuration Manager.

System Center 2012 Configuration Manager does not evaluate the condition for a disabled alert and does not update a disabled alert, even if the state of the alert changes.

Parameters

-Id<String>

Specifies an alert identifier. You can obtain the identifier of an alert by using the **Get-Alert** cmdlet.

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

-InputObject<IResultObject>

Specifies a **CMAAlert** object. To obtain a CMAAlert object, use the **Get-CMAAlert** cmdlet.

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

-Name<String>

Specifies the name of an alert. You can obtain the name of an alert by using **Get-CMAAlert**.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Disable an alert by using alert ID

This command disables an alert that has the Id 16777218.

```
PS C:\> Disable-CMAAlert -Id "16777218"
```

Example 2: Disable an alert by using alert object variable

In this example, the first command gets an alert object that has the Id 16777221 and stores it in the \$AlertObj variable.

The second command disables the alert stored in the \$AlertObj variable.

```
PS C:\> $AlertObj = Get-CMAAlert -Id "16777221"
```

```
PS C:\> Disable-CMAAlert -InputObject $AlertObj
```

Related topics

[Enable-CMAAlert](#)

[Get-CMAAlert](#)

[Remove-CMAAlert](#)

[Set-CMAAlert](#)

[Suspend-CMAAlert](#)

Disable-CMAmtAuditLog

Disable-CMAmtAuditLog

Disables audit logging for Intel AMT-based computers.

Syntax

Parameter Set: SearchByNameMandatory

```
Disable-CMAmtAuditLog -DeviceName <String[]> [-Force] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Disable-CMAmtAuditLog -DeviceId <String[]> [-Force] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Disable-CMAmtAuditLog -Device <IResultObject> [-Force] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Detailed Description

The **Disable-CMAmtAuditLog** cmdlet disables audit logging for Intel Active Management Technology (Intel AMT)-based computers. The audit log records authorized and authenticated out-of-band management activities performed on Intel AMT computers.

You can specify computers by using the Microsoft System Center 2012 Configuration Manager device name or device ID, or you can use the **Get-CMDevice** cmdlet to get a device object. If you want to delete the current log entries, use the **Clear-CMAmtAuditlog** cmdlet.

Parameters

-Device<IResultObject>

Specifies a device object. To obtain a device object, use the **Get-CMDevice** cmdlet.

Aliases	InputObject
Required?	true
Position?	named
Default Value	none

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

-DeviceId<String[]>

Specifies an array of IDs of devices.

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

-DeviceName<String[]>

Specifies an array of names of devices.

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

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Disable audit logging for a device by using an ID

This command disables Intel AMT audit logging for a device that has the ID 16777230.

```
PS C:\> Disable-CMAmtAuditLog -DeviceID "16777230"
```

Example 2: Disable audit logging for named device

This command disables Intel AMT audit logging for a device named Accn023.Contoso.com.

```
PS C:\> Disable-CMAmtAuditLog -DeviceName "Accn023.Contoso.com"
```

Example 3: Disable audit logging by using a variable

The first command gets a device object by using the **Get-CMDevice** command, and stores it in the \$CMD variable.

The second command disables Intel AMT audit logging for the device stored in the \$CMD variable. The command uses the *Force* parameter. Therefore, it does not prompt you for confirmation.

```
PS C:\> $CMD = Get-CMDevice -Name "Accn023.Contoso.com"
```

```
PS C:\> Disable-CMAmtAuditLog -Device $CMD
```

Related topics

[Clear-CMAmtAuditLog](#)

[Enable-CMAmtAuditLog](#)

[Get-CMDevice](#)

Disable-CMBaseline

Disable-CMBaseline

Disables configuration baselines.

Syntax

Parameter Set: SearchByIdMandatory

```
Disable-CMBaseline -Id <String[]> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByNameMandatory

```
Disable-CMBaseline -Name <String[]> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Disable-CMBaseline -InputObject <IResultObject> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Disable-CMBaseline** cmdlet disables one or more configuration baselines in Microsoft System Center 2012 Configuration Manager. After you disable a configuration baseline, System Center 2012 Configuration Manager no longer evaluates it for compliance on client computers.

Parameters

-Id<String[]>

Specifies an array of IDs of configuration baselines.

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

-InputObject<IResultObject>

Specifies a **CMBaseline** object. To obtain a **CMBaseline** object, use the **Get-CMBaseline** cmdlet.

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

-Name<String[]>

Specifies an array of names of configuration baselines.

Aliases	LocalizedDisplayName
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Disable a configuration baseline

This command disables the configuration baseline named BLconfig01.

```
PS C:\> Disable-CMBaseline -Name "BLconfig01"
```

Related topics

[Enable-CMBaseline](#)

[Export-CMBaseline](#)

[Get-CMBaseline](#)

[Import-CMBaseline](#)

[New-CMBaseline](#)

[Remove-CMBaseline](#)

[Set-CMBaseline](#)

[Get-CMBaselineXMLDefinition](#)

[Get-CMBaselineSummarizationSchedule](#)

Disable-CMDriver

Disable-CMDriver

Disables a software driver or a device driver.

Syntax

Parameter Set: SearchByIdMandatory

```
Disable-CMDriver -Id <String[]> [-SecuredScopeNames <String> ] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByNameMandatory

```
Disable-CMDriver -Name <String[]> [-SecuredScopeNames <String> ] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Disable-CMDriver -InputObject <IResultObject> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Disable-CMDriver** cmdlet disables a software driver or a device driver in Microsoft System Center 2012 Configuration Manager.

Parameters

-Id<String[]>

Specifies an array of identifiers for a driver.

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

-InputObject<IResultObject>

Specifies a driver object. To obtain a driver object, use the **Get-CMDriver** object.

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

-Name<String[]>

Specifies an array of names for a driver.

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

-SecuredScopeNames<String>

Specifies the names of security scopes. A security scope name can be Default or the name of a custom-created security scope.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Disable a driver that is specified by its identifier

This command disables a device driver that is specified by its identifier.

```
PS C:\> Disable-CMDriver -Id "16777705"
```

Example 2: Disable a device driver that is specified by its name

This command disables a device driver that is specified by its name.

```
PS C:\> Disable-CMDriver -Name "cdrom.inf"
```

Example 3: Disable a driver that is specified by an input object

The first command uses the **Get-CMDriver** cmdlet to assign the variable \$Driver to the device driver that is specified by using its identifier.

The second command disables the device driver that is specified by the variable \$Driver.

```
PS C:\> $Driver = Get-CMDriver -Id "16777705"  
PS C:\> Disable-CMDriver -InputObject $Driver
```

Related topics

[Enable-CMDriver](#)

[Import-CMDriver](#)

[Get-CMDriver](#)

[Remove-CMDriver](#)

[Set-CMDriver](#)

[Get-CMDriverPackage](#)

Disable-CMProgram

Disable-CMProgram

Disables programs in Configuration Manager packages.

Syntax

Parameter Set: SearchByNameAndNameMandatory

```
Disable-CMProgram -PackageName <String[]> -ProgramName <String[]> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByIdAndNameMandatory

```
Disable-CMProgram -PackageId <String[]> -ProgramName <String[]> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Detailed Description

The **Disable-CMProgram** cmdlet disables one or more programs in Microsoft System Center 2012 Configuration Manager packages. Programs are commands that are associated with a System Center 2012 Configuration Manager package. Programs identify the actions that occur when the client receives the client package. You can associate multiple programs with the same package.

You can disable a program to prevent System Center 2012 Configuration Manager from running it on client computers where it is currently advertised. When you disable a program, System Center 2012 Configuration Manager still sends the program to distribution points and still advertises the Program on client computers, but Configuration Manager does not display or run the program on the client. This behavior is the same that occurs when you disable an advertisement with which the program has been associated.

Parameters

-PackageId<String[]>

Specifies an array of package IDs.

Aliases	none
Required?	true
Position?	named

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

-PackageName<String[]>

Specifies an array of package names.

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

-ProgramName<String[]>

Specifies an array of program names.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Disable a program

This command disables the program named ProgramD02 in the package that has the ID CM400007.

```
PS C:\> Disable-CMProgram -PackageId "CM400007" -ProgramName "ProgramD02"
```

Related topics

[Enable-CMProgram](#)

[Get-CMProgram](#)

[New-CMProgram](#)

[Remove-CMProgram](#)

[Set-CMProgram](#)

[Get-CMPackage](#)

Disable-CMSoftwareMeteringRule

Disable-CMSoftwareMeteringRule

Disables Configuration Manager software metering rules.

Syntax

Parameter Set: SearchByIdMandatory

```
Disable-CMSoftwareMeteringRule -Id <String[]> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByNameMandatory

```
Disable-CMSoftwareMeteringRule -ProductName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Disable-CMSoftwareMeteringRule -InputObject <IResultObject> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Disable-CMSoftwareMeteringRule** cmdlet disables one or more software metering rules in Microsoft System Center 2012 Configuration Manager. If you disable a rule, it does not collect information from clients. You can use the **Enable-CMSoftwareMeteringRule** cmdlet to enable a rule that you previously disabled.

Software metering monitors and collects software usage data from System Center 2012 Configuration Manager clients, such as when clients began using a particular software program and how long users have worked with that software. You can create software metering rules that specify which software to monitor.

You can specify rules that disable software metering rules by ID or by product name, or use the **Get-CMSoftwareMeteringRule** cmdlet. You can use the **Remove-CMSoftwareMeteringRule** to permanently delete a rule.

For more information about software metering in System Center 2012 Configuration Manager, see [Introduction to Software Metering in Configuration Manager](http://go.microsoft.com/fwlink/?LinkId=268432) (<http://go.microsoft.com/fwlink/?LinkId=268432>) on TechNet.

Parameters

-Id<String[]>

Specifies an array of IDs for software metering rules.

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

-InputObject<IResultObject>

Specifies a software metering rule object. To obtain a software metering rule object, use the **Get-SoftwareMeteringRule** cmdlet.

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

-ProductName<String>

Specifies a name for a product that a rule meters.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Disable rules for a specific product

This command disables software metering rules for the product named Accounting Package. There may be more than one rule.

```
PS C:\> Disable-CMSoftwareMeteringRule -ProductName "Accounting Package"
```

Example 2: Disable a specific rule

This command disables a software metering rule that has the specified ID.

```
PS C:\> Disable-CMSoftwareMeteringRule -Id "16777229"
```

Related topics

[Enable-CMSoftwareMeteringRule](#)

[Get-CMSoftwareMeteringRule](#)

[New-CMSoftwareMeteringRule](#)

[Remove-CMSoftwareMeteringRule](#)

[Set-CMSoftwareMeteringRule](#)

Disable-CMSoftwareUpdateAutoDeploymentRule

Disable-CMSoftwareUpdateAutoDeploymentRule

Disables Configuration Manager deployment rules for automatic software updates.

Syntax

Parameter Set: SearchByIdMandatory

```
Disable-CMSoftwareUpdateAutoDeploymentRule -Id <String[]> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByNameMandatory

```
Disable-CMSoftwareUpdateAutoDeploymentRule -Name <String> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Disable-CMSoftwareUpdateAutoDeploymentRule -InputObject <IResultObject> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Detailed Description

The **Disable-CMSoftwareUpdateAutoDeploymentRule** cmdlet disables specified Microsoft System Center 2012 Configuration Manager deployment rules for automatic software updates. While a rule is disabled, it does not run in accordance with its schedule and you cannot run it manually.

System Center 2012 Configuration Manager uses rules to manage automatic deployment of software updates. When a rule runs, System Center 2012 Configuration Manager adds updates that qualify for the rule to a software update group. The System Center 2012 Configuration Manager server downloads content files and copies them to distribution points, and then updates client computers.

You can specify rules to disable by ID or by name, or specify a rule object by using the **Get-CMSoftwareUpdateAutoDeploymentRule** cmdlet. You can use the **Enable-CMSoftwareUpdateAutoDeploymentRule** cmdlet to enable a rule. To remove a rule permanently, use the **Remove-CMSoftwareUpdateAutoDeploymentRule** cmdlet.

Parameters

-Id<String[]>

Specifies an array of IDs for rules for automatic deployment of software updates. This value is the **AutoDeploymentID** property of the deployment rule object.

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

-InputObject<IResultObject>

Specifies a software update automatic deployment rule object. To obtain a deployment rule object, use the **Get-CMSoftwareUpdateAutoDeploymentRule** cmdlet.

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

-Name<String>

Specifies a name of a rule for automatic deployment of software updates.

Aliases	none
Required?	true
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Disable a deployment rule by name

This command disables a rule named Weekly Driver Updates.

```
PS C:\> Disable-CMSoftwareUpdateAutoDeploymentRule -Name "Weekly Driver Updates"
```

Example 2: Disable a deployment rule by ID

This command disables a deployment rule that has the ID 16777217.

```
PS C:\> Disable-CMSoftwareUpdateAutoDeploymentRule -Id "16777217"
```

Example 3: Disable a deployment rule by using a variable

The first command gets a deployment rule that has the specified name and stores it in the \$CMSUADR variable.

The second command disables the rule stored in the variable.

```
PS C:\> $CMSUADR = Get-CMSoftwareUpdateAutoDeploymentRule -Name "Weekly Driver Updates"
```

```
PS C:\> Disable-CMSoftwareUpdateAutoDeploymentRule -InputObject $CMSUADR
```

Related topics

[Enable-CMSoftwareUpdateAutoDeploymentRule](#)

[Get-CMSoftwareUpdateAutoDeploymentRule](#)

[Invoke-CMSoftwareUpdateAutoDeploymentRule](#)

[New-CMSoftwareUpdateAutoDeploymentRule](#)

[Remove-CMSoftwareUpdateAutoDeploymentRule](#)

[Set-CMSoftwareUpdateAutoDeploymentRule](#)

Disable-CMStatusFilterRule

Disable-CMStatusFilterRule

Disables a Configuration Manager filter rule for status messages.

Syntax

Parameter Set: SearchBySiteCodeMandatory

```
Disable-CMStatusFilterRule -Name <String> -SiteCode <String> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByValue

```
Disable-CMStatusFilterRule -InputObject <IResultObject> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Detailed Description

The **Disable-CMStatusFilterRule** cmdlet disables a specified Microsoft System Center 2012 Configuration Manager filter rule for status messages.

Status filter rules specify how System Center 2012 Configuration Manager responds to status messages. Each filter rule contains criteria and actions for status messages. You configure status filter rules for each site, not across all sites.

Use the rule name and site code to specify a rule to disable. You can use the **Enable-CMStatusFilterRule** cmdlet to enable a rule. To remove a rule permanently, use the **Remove-CMStatusFilterRule** cmdlet.

Parameters

-InputObject<IResultObject>

Specifies a status filter rule object to disable. To obtain a status filter rule object, use the **Get-CMStatusFilterRule** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none

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

-Name<String>

Specifies a name of a rule.

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

-SiteCode<String>

Specifies a site code for the Configuration Manager site.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Disable a status filter rule

This command disables a status filter rule that has the specified name in a site that has the site code CM1.

```
PS C:\> Disable-CMStatusFilterRule -Name "Status change to critical" -SiteCode "CM1"
```

Related topics

[Enable-CMStatusFilterRule](#)

[Get-CMStatusFilterRule](#)

[New-CMStatusFilterRule](#)

[Remove-CMStatusFilterRule](#)

[Set-CMStatusFilterRule](#)

Disable-CMTaskSequence

Disable-CMTaskSequence

Disables a Configuration Manager task sequence.

Syntax

Parameter Set: SearchByIdMandatory

```
Disable-CMTaskSequence -TaskSequencePackageId <String[]> [-SecuredScopeNames <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByNameMandatory

```
Disable-CMTaskSequence -Name <String> [-SecuredScopeNames <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Disable-CMTaskSequence -InputObject <IResultObject> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Disable-CMTaskSequence** cmdlet disables a Microsoft System Center 2012 Configuration Manager task sequence. A task sequence includes configuration and operating system deployment settings for a System Center 2012 Configuration Manager client computer. When you disable a task sequence, System Center 2012 Configuration Manager no longer advertises the sequence to clients.

You can specify a name or ID to disable a specific sequence or use the **Get-CMTaskSequence** cmdlet to obtain a sequence to disable. To disable a sequence only if it has a particular security scope, you can specify a security scope along with a name or ID.

You can use the **Enable-CMTaskSequence** cmdlet to enable a sequence later. To remove a task sequence completely, use the **Remove-CMTaskSequence** cmdlet.

Parameters

-InputObject<IResultObject>

Specifies a task sequence object. To obtain a task sequence object, use the **Get-CMTaskSequence** cmdlet.

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

-Name<String>

Specifies a name of a task sequence.

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

-SecuredScopeNames<String>

Specifies a name of a security scope.

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

-TaskSequencePackageId<String[]>

Specifies an array of IDs of task sequences.

Aliases	PackageId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Disable a specific task sequence

This command disables a task sequence that has an ID of CM100008.

```
PS C:\> Disable-CMTaskSequence -Id "CM100008"
```

Example 2: Disable a task sequence by using a variable

The first command gets the task sequence that has an ID of CM100007 and stores it in the \$CMTS variable.

The second command disables the task sequence stored in \$CMTS.

```
PS C:\> $CMTS = Get-CMTaskSequence -Id CM100007
```

```
PS C:\> Disable-CMTaskSequence -InputObject $CMTS
```

Related topics

[Enable-CMTaskSequence](#)

[Export-CMTaskSequence](#)

[Get-CMTaskSequence](#)

[Import-CMTaskSequence](#)

[New-CMTaskSequence](#)

[Remove-CMTaskSequence](#)

[Set-CMTaskSequence](#)

Enable-CMAAlert

Enable-CMAAlert

Enables Configuration Manager alerts.

Syntax

Parameter Set: SearchByIdMandatory

```
Enable-CMAAlert -Id <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByNameMandatory

```
Enable-CMAAlert -Name <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Enable-CMAAlert -InputObject <IResultObject> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Enable-CMAAlert** cmdlet enables one or more Microsoft System Center 2012 Configuration Manager alerts.

Parameters

-Id<String>

Specifies an alert identifier. You can obtain the identifier of an alert by using the **Get-Alert** cmdlet.

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

-InputObject<IResultObject>

Specifies a **CMAAlert** object. To obtain a **CMAAlert** object, use the **Get-CMAAlert** cmdlet.

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

-Name<String>

Specifies an alert name. You can obtain the name of an alert by using **Get-CMAAlert**.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Enable an alert by using alert ID

This command enables an alert that has the Id 16777223.

```
PS C:\> Enable-CMAAlert -Id "16777223"
```

Example 2: Enable an alert by using an alert object variable

In this example, the first command gets the alert object that has the ID 16777218 and stores it in the \$AlertObj variable.

The second command enables the alert stored in the \$AlertObj variable.

```
PS C:\> $AlertObj = Get-CMAAlert -Id "16777218"  
PS C:\> Enable-CMAAlert -InputObject $AlertObj
```

Related topics

[Get-CMAAlert](#)

[Remove-CMAAlert](#)

[Set-CMAAlert](#)

[Suspend-CMAAlert](#)

[Disable-CMAAlert](#)

Enable-CMAmtAuditLog

Enable-CMAmtAuditLog

Enables audit logging for Intel AMT-based computers.

Syntax

Parameter Set: SearchByNameMandatory

```
Enable-CMAmtAuditLog -DeviceName <String[]> [-Force] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Enable-CMAmtAuditLog -DeviceId <String[]> [-Force] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Enable-CMAmtAuditLog -Device <IResultObject> [-Force] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Detailed Description

The **Enable-CMAmtAuditLog** cmdlet enables audit logging for Intel Active Management Technology (Intel AMT)-based computers. The audit log records authorized and authenticated out-of-band management activities performed on Intel AMT computers.

You can specify computers by using the Microsoft System Center 2012 Configuration Manager device name or device ID, or you can use the **Get-CMDevice** cmdlet to get a device object.

Parameters

-Device<IResultObject>

Specifies a device object in Configuration Manager. To obtain a device object, use the **Get-CMDevice** cmdlet.

Aliases	InputObject
Required?	true
Position?	named
Default Value	none

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

-DeviceId<String[]>

Specifies an array of IDs of devices.

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

-DeviceName<String[]>

Specifies an array of names of devices.

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

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Enable audit logging for a device by using an ID

This command enables Intel AMT audit logging for a device that has the ID 16777230.

```
PS C:\> Enable-CMAmtAuditLog -DeviceID "16777230"
```

Example 2: Enable audit logging for a named device

This command enables Intel AMT audit logging for a device named Accn023.Contoso.com.

```
PS C:\> Enable-CMAmtAuditLog -DeviceName "Accn023.Contoso.com"
```

Example 3: Enable audit logging by using a variable

The first command gets a device object by using the **Get-CMDevice** command, and stores it in the \$CMD variable.

The second command enables Intel AMT audit logging for the device stored in the \$CMD variable. The command uses the *Force* parameter. Therefore, the command does not prompt you for confirmation.

```
PS C:\> $CMD = Get-CMDevice -Name "Accn023.Contoso.com"
```

```
PS C:\> Enable-CMAmtAuditLog -Device $CMD -Force
```

Related topics

[Clear-CMAmtAuditLog](#)

[Disable-CMAmtAuditLog](#)

[Get-CMDevice](#)

Enable-CMAutomaticAMTProvisioning

Enable-CMAutomaticAMTProvisioning

Enables automatic provisioning of AMT-based computers.

Syntax

Parameter Set: SearchByNameMandatory

```
Enable-CMAutomaticAMTProvisioning -DeviceName <String[]> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Enable-CMAutomaticAMTProvisioning -DeviceId <String[]> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Enable-CMAutomaticAMTProvisioning -Device <IResultObject> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Detailed Description

The **Enable-CMAutomaticAMTProvisioning** cmdlet enables automatic provisioning of Intel Active Management Technology (AMT)-based computers for Microsoft System Center 2012 Configuration Manager. After System Center 2012 Configuration Manager identifies computers with an AMT management controller from an out of band service point, you can use this cmdlet to enable automatic provision of the computers. An out of band service point is a site system role that provisions and configures Intel Active Management Technology (AMT)-based computers for System Center 2012 Configuration Manager.

Parameters

-Device<IResultObject>

Specifies a device object. To obtain a **CMDevice** object, use the **Get-CMDevice** cmdlet.

Aliases	none
Required?	true
Position?	named

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

-DeviceId<String[]>

Specifies an array of device IDs.

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

-DeviceName<String[]>

Specifies an array of device names.

Aliases	Name
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Enable automatic provisioning of an AMT-based computers

This command enables automatic provisioning of the device named CMdiv-west03.

```
PS C:\> Enable-CMAutomaticAmtProvisioning -DeviceName "CMdiv-west03"
```

Related topics

[Get-CMAutomaticAmtProvisioningStatus](#)

[Get-CMDevice](#)

Enable-CMBaseline

Enable-CMBaseline

Enables configuration baselines.

Syntax

Parameter Set: SearchByIdMandatory

```
Enable-CMBaseline -Id <String[]> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByNameMandatory

```
Enable-CMBaseline -Name <String[]> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Enable-CMBaseline -InputObject <IResultObject> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Enable-CMBaseline** cmdlet enables one or more configuration baselines for compliance monitoring.

Parameters

-Id<String[]>

Specifies an array of IDs of configuration baselines.

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

-InputObject<IResultObject>

Specifies a **CMBaseline** object. To obtain a **CMBaseline** object, use the **Get-CMBaseline** cmdlet.

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

-Name<String[]>

Specifies an array of names of configuration baselines.

Aliases	LocalizedDisplayName
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Enable a configuration baseline

This command enables the configuration baseline named BLconfig01.

```
PS C:\> Enable-CMBaseline -Name "BLconfig01"
```

Related topics

[Disable-CMBaseline](#)

[Export-CMBaseline](#)

[Get-CMBaseline](#)

[Import-CMBaseline](#)

[New-CMBaseline](#)

[Remove-CMBaseline](#)

[Set-CMBaseline](#)

Enable-CMDriver

Enable-CMDriver

Enables a software driver or a device driver.

Syntax

Parameter Set: SearchByIdMandatory

```
Enable-CMDriver -Id <String[]> [-SecuredScopeNames <String> ] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByNameMandatory

```
Enable-CMDriver -Name <String[]> [-SecuredScopeNames <String> ] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Enable-CMDriver -InputObject <IResultObject> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Enable-CMDriver** cmdlet enables a software driver or device driver in Microsoft System Center 2012 Configuration Manager.

Parameters

-Id<String[]>

Specifies an array of identifiers for a driver.

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

-InputObject<IResultObject>

Specifies a driver object. To obtain a driver object, use the **Get-CMDriver** cmdlet.

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

-Name<String[]>

Specifies an array of names for a driver.

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

-SecuredScopeNames<String>

Specifies the names of security scopes. A security scope name can be Default or the name of a custom-created security scope.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Enable a driver that is specified by its identifier

This command enables a device driver that is specified by its identifier.

```
PS C:\> Enable-CMDriver -Id "16777705"
```

Example 2: Enable a device driver that is specified by its name

This command enables a device driver that is specified by its name.

```
PS C:\> Enable-CMDriver -Name "cdrom.inf"
```

Example 3: Enable a driver that is retrieved by its ID

The first command uses the **Get-CMDriver** cmdlet to get the device driver object with the ID of 16777705 and stores the object in the \$Driver variable.

The second command enables the device driver that is stored in \$Driver.

```
PS C:\> $Driver =Get-CMDriver -Id "16777705"
```

```
PS C:\> Enable-CMDriver -InputObject $Driver
```

Related topics

[Disable-CMDriver](#)

[Get-CMDriver](#)

[Remove-CMDriver](#)

[Get-CMDriverPackage](#)

Enable-CMProgram

Enable-CMProgram

Enables programs in Configuration Manager packages.

Syntax

Parameter Set: SearchByNameAndNameMandatory

```
Enable-CMProgram -PackageName <String[]> -ProgramName <String[]> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByIdAndNameMandatory

```
Enable-CMProgram -PackageId <String[]> -ProgramName <String[]> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Detailed Description

The **Enable-CMProgram** cmdlet enables one or more programs in Microsoft System Center 2012 Configuration Manager packages. You can enable a program that has been disabled in order to resume availability of that program to client computers. You can also enable a program by enabling an advertisement that you used to disable it.

Programs are commands that are associated with a System Center 2012 Configuration Manager package. Programs identify the actions that occur when the client receives the client package. You can associate multiple programs with the same package.

Parameters

-PackageId<String[]>

Specifies an array of package IDs.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

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

-PackageName<String[]>

Specifies an array of package names.

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

-ProgramName<String[]>

Specifies an array of program names.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Enable a program

This command enables the program named ProgramD02 in the package that has the ID CM400007.

```
PS C:\> Enable-CMProgram -PackageId "CM400007" -ProgramName "ProgramD02"
```

Related topics

[Disable-CMProgram](#)

[Get-CMProgram](#)

[New-CMProgram](#)

[Remove-CMProgram](#)

[Set-CMProgram](#)

[Get-CMPackage](#)

Enable-CMSoftwareMeteringRule

Enable-CMSoftwareMeteringRule

Enables Configuration Manager software metering rules.

Syntax

Parameter Set: SearchByIdMandatory

```
Enable-CMSoftwareMeteringRule -Id <String[]> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByNameMandatory

```
Enable-CMSoftwareMeteringRule -ProductName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Enable-CMSoftwareMeteringRule -InputObject <IResultObject> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Enable-CMSoftwareMeteringRule** cmdlet enables one or more software metering rules in Microsoft System Center 2012 Configuration Manager. You can enable a rule that you previously disabled by using the **Disable-CMSoftwareMeteringRule** cmdlet. When System Center 2012 Configuration Manager automatically creates software metering rules, it creates them as disabled.

Software metering monitors and collects software usage data from System Center 2012 Configuration Manager clients, such as when clients began using a particular software program and how long users have worked with that software. You can create software metering rules that specify which software to monitor.

You can specify rules that enable software metering rules by ID or by product name, or use the **Get-CMSoftwareMeteringRule** cmdlet.

For more information about software metering in System Center 2012 Configuration Manager, see [Introduction to Software Metering in Configuration Manager](http://go.microsoft.com/fwlink/?LinkId=268432) (<http://go.microsoft.com/fwlink/?LinkId=268432>) on TechNet.

Parameters

-Id<String[]>

Specifies an array of IDs for software metering rules.

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

-InputObject<IResultObject>

Specifies a software metering rule object. To obtain a software metering rule object, use the **Get-SoftwareMeteringRule** cmdlet.

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

-ProductName<String>

Specifies a name for a product that a rule meters.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Enable rules for a specific product

This command enables software metering rules for a product named Accounting Package. There may be more than one rule. If you previously disabled some rules for this product, but not all, the cmdlet does not inform you that some rules were already enabled.

```
PS C:\> Enable-CMSoftwareMeteringRule -ProductName "Accounting Package"
```

Example 2: Enable a specific rule

This command enables a software metering rule that has the specified ID.

```
PS C:\> Enable-CMSoftwareMeteringRule -Id "16777229"
```

Related topics

[Disable-CMSoftwareMeteringRule](#)

[Get-CMSoftwareMeteringRule](#)

[New-CMSoftwareMeteringRule](#)

[Remove-CMSoftwareMeteringRule](#)

[Set-CMSoftwareMeteringRule](#)

Enable-CMSoftwareUpdateAutoDeploymentRule

Enable-CMSoftwareUpdateAutoDeploymentRule

Enables Configuration Manager deployment rules for automatic software updates.

Syntax

Parameter Set: SearchByIdMandatory

```
Enable-CMSoftwareUpdateAutoDeploymentRule -Id <String[]> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByNameMandatory

```
Enable-CMSoftwareUpdateAutoDeploymentRule -Name <String> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Enable-CMSoftwareUpdateAutoDeploymentRule -InputObject <IResultObject> [-Confirm] [-WhatIf]  
[ <CommonParameters>]
```

Detailed Description

The **Enable-CMSoftwareUpdateAutoDeploymentRule** cmdlet enables specified Microsoft System Center 2012 Configuration Manager deployment rules for automatic software updates. While a rule is disabled, it does not run in accordance with its schedule and you cannot run it manually.

System Center 2012 Configuration Manager uses rules to manage automatic deployment of software updates. When a rule runs, System Center 2012 Configuration Manager adds updates that qualify for the rule to a software update group. The System Center 2012 Configuration Manager server downloads content files and copies them to distribution points, and then updates client computers.

You can specify rules to enable by ID or by name, or specify a rule object by using the **Get-CMSoftwareUpdateAutoDeploymentRule** cmdlet. You can use the **Disable-CMSoftwareUpdateAutoDeploymentRule** cmdlet to disable a rule. To remove a rule permanently, use the **Remove-CMSoftwareUpdateAutoDeploymentRule** cmdlet.

Parameters

-Id<String[]>

Specifies an array of IDs for rules for automatic deployment of software updates. This value is the **AutoDeploymentID** property of the deployment rule object.

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

-InputObject<IResultObject>

Specifies a software update automatic deployment rule object. To obtain a deployment rule object, use the **Get-CMSoftwareUpdateAutoDeploymentRule** cmdlet.

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

-Name<String>

Specifies a name of a rule for automatic deployment of software updates.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Enable a deployment rule by name

This command enables a rule named Weekly Driver Updates.

```
PS C:\> Enable-CMSoftwareUpdateAutoDeploymentRule -Name "Weekly Driver Updates"
```

Example 2: Enable a deployment rule by ID

This command enables a deployment rule that has the ID 16777217.

```
PS C:\> Enable-CMSoftwareUpdateAutoDeploymentRule -Id "16777217"
```

Example 3: Enable a deployment rule by using a variable

The first command gets a deployment rule that has the specified name and stores it in the \$CMSUADR variable.

The second command enables the rule stored in the variable.

```
PS C:\> $CMSUADR = Get-CMSoftwareUpdateAutoDeploymentRule -Name "Weekly Driver Updates"
PS C:\> Enable-CMSoftwareUpdateAutoDeploymentRule -InputObject $CMSUADR
```

Related topics

[Disable-CMSoftwareUpdateAutoDeploymentRule](#)

[Get-CMSoftwareUpdateAutoDeploymentRule](#)

[Invoke-CMSoftwareUpdateAutoDeploymentRule](#)

[New-CMSoftwareUpdateAutoDeploymentRule](#)

[Remove-CMSoftwareUpdateAutoDeploymentRule](#)

[Set-CMSoftwareUpdateAutoDeploymentRule](#)

Enable-CMStatusFilterRule

Enable-CMStatusFilterRule

Enables a Configuration Manager filter rule for status messages.

Syntax

Parameter Set: SearchBySiteCodeMandatory

```
Enable-CMStatusFilterRule -Name <String> -SiteCode <String> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByValue

```
Enable-CMStatusFilterRule -InputObject <IResultObject> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Detailed Description

The **Enable-CMStatusFilterRule** cmdlet enables a specified Microsoft System Center 2012 Configuration Manager filter rule for status messages.

Status filter rules specify how System Center 2012 Configuration Manager responds to status messages. Each filter rule contains criteria and actions for status messages. You configure status filter rules for each site, not across all sites.

Use the rule name and site code to specify a rule to enable. You can use the **Disable-CMStatusFilterRule** cmdlet to disable a rule. To remove a rule permanently, use the **Remove-CMStatusFilterRule** cmdlet.

Parameters

-InputObject<IResultObject>

Specifies a status filter rule object to enable. To obtain a status filter rule object, use the **Get-CMStatusFilterRule** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none

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

-Name<String>

Specifies a name of a rule.

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

-SiteCode<String>

Specifies a site code for the Configuration Manager site.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Enable a status filter rule

This command enables a status filter rule that has the specified name in a site that has the site code CM1.

```
PS C:\> Enable-CMStatusFilterRule -Name "Status change to critical" -SiteCode "CM1"
```

Related topics

[Disable-CMStatusFilterRule](#)

[Get-CMStatusFilterRule](#)

[New-CMStatusFilterRule](#)

[Remove-CMStatusFilterRule](#)

[Set-CMStatusFilterRule](#)

Enable-CMTaskSequence

Enable-CMTaskSequence

Enables a Configuration Manager task sequence.

Syntax

Parameter Set: SearchByIdMandatory

```
Enable-CMTaskSequence -TaskSequencePackageId <String[]> [-SecuredScopeNames <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByNameMandatory

```
Enable-CMTaskSequence -Name <String> [-SecuredScopeNames <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Enable-CMTaskSequence -InputObject <IResultObject> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Enable-CMTaskSequence** cmdlet enables a Microsoft System Center 2012 Configuration Manager task sequence. A task sequence includes configuration and operating system deployment settings for a System Center 2012 Configuration Manager client computer. You can use the **Disable-CMTaskSequence** cmdlet to disable a task sequence. When you disable a task sequence, System Center 2012 Configuration Manager no longer advertises the sequence to clients.

You can specify a name or ID to enable a specific sequence or use the **Get-CMTaskSequence** cmdlet to obtain a sequence to enable. To enable a sequence only if it has a particular security scope, you can specify a security scope along with a name or ID.

Parameters

-InputObject<IResultObject>

Specifies a task sequence object. To obtain a task sequence object, use the **Get-CMTaskSequence** cmdlet.

Aliases	none
Required?	true

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies a name for a task sequence.

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

-SecuredScopeNames<String>

Specifies a name of a security scope.

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

-TaskSequencePackageId<String[]>

Specifies an array of IDs of task sequences.

Aliases	PackageId
Required?	true

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Enable a specific task sequence

This command enables the task sequence that has an ID of CM100008.

```
PS C:\> Enable-CMTaskSequence -Id "CM100008"
```

Example 2: Enable a task sequence by using a variable

The first command uses the **Get-CMTaskSequence** cmdlet to get the task sequence that has an ID of CM100007 and stores it in the \$CMTS variable.

The second command enables the sequence stored in \$CMTS.

```
PS C:\> $CMTS = Get-CMTaskSequence -Id "CM100007"
```

```
PS C:\> Enable-CMTaskSequence -InputObject $CMTS
```

Related topics

[Disable-CMTaskSequence](#)

[Export-CMTaskSequence](#)

[Get-CMTaskSequence](#)

[Import-CMTaskSequence](#)

[New-CMTaskSequence](#)

[Remove-CMTaskSequence](#)

[Set-CMTaskSequence](#)

Export-CMAntimalwarePolicy

Export-CMAntimalwarePolicy

Exports an antimalware policy for Endpoint Protection.

Syntax

Parameter Set: ExportByNameMandatory

```
Export-CMAntimalwarePolicy -ExportFilePath <String> -Name <String> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: ExportByIdMandatory

```
Export-CMAntimalwarePolicy -ExportFilePath <String> -Id <String> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: ExportByValueMandatory

```
Export-CMAntimalwarePolicy -ExportFilePath <String> -InputObject <IResultObject> [-Confirm]  
[-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Export-CMAntiMalwarePolicy** cmdlet exports an antimalware policy for System Center 2012 Endpoint Protection. Policies include information about the scan schedule, the types of files and folders to scan, and the actions to take when a scan detects malware.

Parameters

-ExportFilePath<String>

Specifies a full file path to which you export the policy.

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

-Id<String>

Specifies the policy that you export by using an ID.

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

-InputObject<IResultObject>

Specifies the policy that you export by using an antimalware policy object. To obtain an antimalware policy object, use the **Get-CMAntiMalwarePolicy** cmdlet.

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

-Name<String>

Specifies the policy that you export by using a name.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

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

Example 1: Export an antimalware policy by using an ID

This command exports an antimalware policy that has the ID 16777568 to the file named AMPolicy05.xml in the folder named C:\Exports\Policy.

```
PS C:\> Export-CMAntiMalwarePolicy -Id "16777568" -Path "C:\Exports\Policy\AMPolicy05.xml"
```

Example 2: Export an antimalware policy by using a variable

The first command gets the antimalware policy object that has the ID 16777568 and assigns it to the variable named \$AMPol01.

The second command exports the policy in the in the \$AMPol01 variable to the file named AMPolicy05.xml in the folder named C:\Exports\Policy.

```
PS C:\> $AMPol01 = Get-CMAntiMalwarePolicy -Id "16777217"  
PS C:\> Export-CMAntiMalwarePolicy -InputObject $AMPol01 -Path  
"C:\Exports\Policy\AMPolicy05.xml"
```

Related topics

[Get-CMAntiMalwarePolicy](#)

[Import-CMAntimalwarePolicy](#)

[Merge-CMAntimalwarePolicy](#)

[New-CMAntimalwarePolicy](#)

[Remove-CMAntiMalwarePolicy](#)

[Set-CMAntiMalwarePolicy](#)

Export-CMApplication

Export-CMApplication

Exports an application from Configuration Manager.

Syntax

Parameter Set: SearchByNameMandatory

```
Export-CMApplication -Name <String[]> -Path <String> [-Comment <String> ] [-IgnoreRelated] [-OmitContent] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Export-CMApplication -Id <String[]> -Path <String> [-Comment <String> ] [-IgnoreRelated] [-OmitContent] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Export-CMApplication -InputObject <IResultObject> -Path <String> [-Comment <String> ] [-IgnoreRelated] [-OmitContent] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Export-CMApplication** cmdlet exports an application to a file, and copies the content to a separate folder if *OmitContent* is false. Specify a file path to the location where you want to export the application.

Parameters

-Comment<String>

Specifies a comment for the exported application.

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

-Id<String[]>

Specifies an array of IDs for the exported application.

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

-IgnoreRelated

Indicates that related objects, such as application dependencies, superseded applications, or related categories and global conditions, are exported.

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

-InputObject<IResultObject>

Specifies an application object. To obtain an application object, use the **Get-CMApplication** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

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

-Name<String[]>

Specifies an array of names for the exported application.

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

-OmitContent

Indicates that the cmdlet exports related content to a separate folder in the same location as the .zip file.

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

-Path<String>

Specifies a path of the package. The package file has a .zip extension.

Aliases	none
Required?	true
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Export an application

This command exports the Configuration Manager application named BL_App_01 to the file named Test.zip.

```
PS C:\> Export-CMApplication -Name "BL_App_01" -Path "\\Contoso01\CM\Toolbox\Test.zip"
```

Related topics

[Get-CMApplication](#)

[Import-CMApplication](#)

[New-CMApplication](#)

[Remove-CMApplication](#)

[Resume-CMApplication](#)

[Set-CMApplication](#)

[Suspend-CMApplication](#)

Export-CMBaseline

Export-CMBaseline

Exports configuration baselines.

Syntax

Parameter Set: SearchByNameMandatory

```
Export-CMBaseline -Name <String[]> -Path <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Export-CMBaseline -Id <String[]> -Path <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Export-CMBaseline -InputObject <IResultObject> -Path <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Export-CMBaseline** cmdlet exports configuration baselines in a cabinet (.cab) file format from a Microsoft System Center 2012 Configuration Manager site. You can then import it to the same or a different System Center 2012 Configuration Manager site. Configuration data is converted to desired configuration management (DCM) Digest.

Parameters

-Id<String[]>

Specifies an array of IDs of configuration baselines.

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

-InputObject<IResultObject>

Specifies a **CMBaseline** object. To obtain a **CMBaseline** object, use the **Get-CMBaseline** cmdlet.

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

-Name<String[]>

Specifies an array of names of configuration baselines.

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

-Path<String>

Specifies the full path of the cabinet (.cab) file.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Export a baseline

This command exports the configuration baseline named BLconfig01 to the file named BaselineW2K8.cab.

```
PS C:\> Export-CMBaseline -Name "BLconfig01" -Path "\\Contoso01\CM\Toolbox\BaselineW2K8.cab"
```

Related topics

[Disable-CMBaseline](#)

[Enable-CMBaseline](#)

[Get-CMBaseline](#)

[Import-CMBaseline](#)

[New-CMBaseline](#)

[Remove-CMBaseline](#)

[Set-CMBaseline](#)

[Get-CMBaselineXMLDefinition](#)

[Get-CMBaselineSummarizationSchedule](#)

Export-CMConfigurationItem

Export-CMConfigurationItem

Saves a Configuration Manager configuration item to a file.

Syntax

Parameter Set: SearchByNameMandatory

```
Export-CMConfigurationItem -Name <String[]> -Path <String> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Export-CMConfigurationItem -Id <String[]> -Path <String> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Export-CMConfigurationItem -InputObject <IResultObject> -Path <String> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Detailed Description

The **Export-CMConfigurationItem** cmdlet saves a Microsoft System Center 2012 Configuration Manager configuration item to a specified .cab file. You can specify items by ID, name, or by use of the **Get-CMConfigurationItem** cmdlet.

Configuration items contain one or more settings, along with compliance rules. Items usually define a unit of configuration you want to. For more information about configuration items, see [Introduction to Compliance Settings in Configuration Manager](http://go.microsoft.com/fwlink/?LinkId=211014) (http://go.microsoft.com/fwlink/?LinkId=211014) on TechNet.

Parameters

-Id<String[]>

Specifies an array of identifiers for one or more configuration items. You can use a comma separated list.

Aliases	CiId
Required?	true

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a configuration item object. To obtain a configuration item object, you can use the **Get-
CMConfigurationItem** cmdlet.

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

-Name<String[]>

Specifies an array of names of configuration items. You can use a comma separated list.

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

-Path<String>

Specifies full file path for an export file.

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

Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Export an item using an ID

This command exports a configuration item with the specified identifier to the specified file.

```
PS C:\> Export-CMConfigurationItem -Id "16777568" -Path "C:\Exports\CI16777568.cab"
```

Example 2: Export an item using a name

This command exports a configuration item named ConfigItem76 to the specified file.

```
PS C:\> Export-CMConfigurationItem -Name "ConfigItem76" -Path  
"C:\Exports\CIConfigItem76.cab"
```

Example 3: Export an item using a variable

The first command gets a configuration item with the specified identifier and stores it in the \$CIObj variable.

The second command exports the item in the \$CIObj variable.

```
PS C:\> $CIObj=Get-CMConfigurationItem -Id "16777568"  
PS C:\> Export-CMConfigurationItem -InputObject $CIObj -Path "C:\Exports\CI16777568.cab"
```

Related topics

[Get-CMConfigurationItem](#)

[Get-CMConfigurationItemXMLDefinition](#)

[Import-CMConfigurationItem](#)

[New-CMConfigurationItem](#)

[Remove-CMConfigurationItem](#)

[Set-CMConfigurationItem](#)

[Get-CMConfigurationItemHistory](#)

Export-CMDeviceCollection

Export-CMDeviceCollection

Exports a device collection.

Syntax

Parameter Set: SearchByNameMandatory

```
Export-CMDeviceCollection -ExportFilePath <String> -Name <String> [-ExportComment <String> ]  
[-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Export-CMDeviceCollection -CollectionId <String> -ExportFilePath <String> [-ExportComment  
<String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Export-CMDeviceCollection -ExportFilePath <String> -InputObject <IResultObject> [-  
ExportComment <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Export-CMDeviceCollection** cmdlet exports a device collection. This cmdlet exports a device collection from the site database to a Managed Object Format (MOF) file that you can archive or use at another Microsoft System Center 2012 Configuration Manager site.

Parameters

-CollectionId<String>

Specifies the ID of a device collection.

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

-ExportComment<String>

Specifies a comment for the exported device collection.

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

-ExportFilePath<String>

Specifies the full path for the export file.

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

-InputObject<IResultObject>

Specifies a device collection object. To obtain a **CMDeviceCollection** object, use the **Get-
CMDeviceCollection** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

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

-Name<String>

Specifies the name of a device collection.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Export a device collection

This command exports the device collection named Windows 7 to the MOF file named Win7Devices.

```
PS C:\> Export-CMDeviceCollection -ExportFilePath "\\Contoso01\Export\Win7Devices" -Name "Windows 7"
```

Related topics

[Get-CMDeviceCollection](#)

[Import-CMDeviceCollection](#)

[New-CMDeviceCollection](#)

[Remove-CMDeviceCollection](#)

[Set-CMDeviceCollection](#)

Export-CMDriverPackage

Export-CMDriverPackage

Exports driver packages.

Syntax

Parameter Set: SearchPackageByNameMandatory

```
Export-CMDriverPackage -ExportFilePath <String> -Name <String> [-Comments <String> ] [-WithContent <Boolean> ] [-WithDependence <Boolean> ] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Export-CMDriverPackage -ExportFilePath <String> -InputObject <IResultObject> [-Comments <String> ] [-WithContent <Boolean> ] [-WithDependence <Boolean> ] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchPackageByIdMandatory

```
Export-CMDriverPackage -ExportFilePath <String> -Id <String[]> [-Comments <String> ] [-WithContent <Boolean> ] [-WithDependence <Boolean> ] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Detailed Description

The **Export-CMDriverPackage** cmdlet exports one or more driver packages to a .zip file.

Parameters

-Comments<String>

Specifies a comment for the exported driver packages.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

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

-ExportFilePath<String>

Specifies the full path for the export file.

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

-Id<String[]>

Specifies an array of IDs of driver packages.

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

-InputObject<IResultObject>

Specifies a driver package object. To obtain a **CMDriverPackage** object, use the [Get-CMDriverPackage](#) cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none

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

-Name<String>

Specifies the name of a driver package.

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

-WithContent<Boolean>

Specifies whether to export the content files for the driver packages and drivers.

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

-WithDependence<Boolean>

Specifies whether to export all drivers associated with the driver package.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Export a driver package

This command exports the driver package named DrvPkg01 to the export file DriverPackage01.zip.

```
PS C:\> Export-CMDriverPackage -Name "DrvPkg01" -ExportFilePath  
"\\Contoso02\DriverPackages\DriverPackage01.zip"
```

Related topics

[Get-CMDriverPackage](#)

[Import-CMDriverPackage](#)

[New-CMDriverPackage](#)

[Remove-CMDriverPackage](#)

[Set-CMDriverPackage](#)

Export-CMPackage

Export-CMPackage

Exports a Configuration Manager package.

Syntax

Parameter Set: SearchPackageByNameMandatory

```
Export-CMPackage -ExportFilePath <String> -Name <String> [-Comments <String> ] [-WithContent <Boolean> ] [-WithDependence <Boolean> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValue

```
Export-CMPackage -ExportFilePath <String> -InputObject <IResultObject> [-Comments <String> ] [-WithContent <Boolean> ] [-WithDependence <Boolean> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchPackageByIdMandatory

```
Export-CMPackage -ExportFilePath <String> -Id <String[]> [-Comments <String> ] [-WithContent <Boolean> ] [-WithDependence <Boolean> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Export-CMPackage** cmdlet exports a Microsoft System Center 2012 Configuration Manager package. You can use this cmdlet in System Center 2012 Configuration Manager to create a package of collections, queries, or reports and then export that package so that you can later deploy these items to a different location.

Parameters

-Comments<String>

Specifies a comment to include in the package.

Aliases	none
Required?	false
Position?	named
Default Value	none

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

-ExportFilePath<String>

Specifies the Universal Naming Convention (UNC) path to which you export the package. This path must end with the filename extension .zip.

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

-Id<String[]>

Specifies an array of IDs for packages.

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

-InputObject<IResultObject>

Specifies the package that you export by using a Configuration Manager package object. To obtain a package object, use the **Get-CMPackage** cmdlet.

Aliases	none
Required?	true

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies a name of a package.

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

-WithContent<Boolean>

Indicates whether to include content in the package.

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

-WithDependence<Boolean>

Indicates whether to include dependencies in the package.

Aliases	none
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Export a package by using an ID

This command exports a package that has the ID ST120001 to the output path \\Deploy01\ExportPackages.

```
PS C:\> Export-CMPackage -Id "ST120001" -ExportFilePath "\\Deploy01\ExportPackages"
```

Example 2: Export a package by using a variable

The first command gets the package that has the ID ST120001 and stores it in the variable \$DeplObj.

The second command exports the package to the path \\Deploy01\ExportPackages by using the \$DeplObj variable.

```
PS C:\> $DeplObj = Get-CMPackage -Id "ST120001"  
PS C:\> Export-CMPackage - "ST120001" -ExportFilePath"\\Deploy01\ExportPackages" -  
InputObject $DeplObj
```

Related topics

[Import-CMPackage](#)

[Get-CMPackage](#)

[New-CMPackage](#)

[Remove-CMPackage](#)

[Set-CMPackage](#)

Export-CMSecurityRole

Export-CMSecurityRole

Exports a security role to an XML file.

Syntax

Parameter Set: Default

```
Export-CMSecurityRole -Path <String> -RoleId <String> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Detailed Description

The **Export-CMSecurityRole** cmdlet exports a security role configuration from Microsoft System Center 2012 Configuration Manager to an XML file.

Parameters

-Path<String>

Specifies the path to which you export the XML file for the security role.

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

-RoleId<String>

Specifies the ID of a role.

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

Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Export a security role

This command exports the security role named SMS000CR to the file named Security_Manager.

```
PS C:\> Export-CMSecurityRole -Path "\\Contoso01\Export\Sec_Roles\Security_Manager" -RoleId "SMS000CR"
```

Related topics

[Get-CMSecurityRole](#)

[Set-CMSecurityRole](#)

[Copy-CMSecurityRole](#)

[Import-CMSecurityRole](#)

[Remove-CMSecurityRoleFromAdministrativeUser](#)

[Remove-CMSecurityRole](#)

Export-CMTaskSequence

Export-CMTaskSequence

Exports a Configuration Manager task sequence.

Syntax

Parameter Set: SearchPackageByNameMandatory

```
Export-CMTaskSequence -ExportFilePath <String> -Name <String> [-Comments <String> ] [-WithContent <Boolean> ] [-WithDependence <Boolean> ] [-Confirm] [-WhatIf] [<CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Export-CMTaskSequence -ExportFilePath <String> -InputObject <IResultObject> [-Comments <String> ] [-WithContent <Boolean> ] [-WithDependence <Boolean> ] [-Confirm] [-WhatIf] [<CommonParameters>]
```

Parameter Set: SearchPackageByIdMandatory

```
Export-CMTaskSequence -ExportFilePath <String> -TaskSequencePackageId <String[]> [-Comments <String> ] [-WithContent <Boolean> ] [-WithDependence <Boolean> ] [-Confirm] [-WhatIf] [<CommonParameters>]
```

Detailed Description

The **Export-CMTaskSequence** cmdlet exports a Microsoft System Center 2012 Configuration Manager task sequence to a .zip file.

Parameters

-Comments<String>

Specifies a comment in Configuration Manager.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

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

-ExportFilePath<String>

Specifies a path to the exported ZIP file in Configuration Manager.

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

-InputObject<IResultObject>

Specifies a task sequence object. To obtain a task sequence object, use the **Get-CMTaskSequence** cmdlet.

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

-Name<String>

Specifies a name for the task sequence in Configuration Manager.

Aliases	none
Required?	true
Position?	named
Default Value	none

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

-TaskSequencePackageId<String[]>

Specifies an array of IDs of task sequences.

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

-WithContent<Boolean>

Indicates whether to include content associated with the task sequence in the export ZIP file. If you specify a value for \$True, the cmdlet copies the content from the package source to the export location, and the Import Task Sequence Wizard uses the import path as the new package source location.

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

-WithDependence<Boolean>

Indicates whether to include dependencies in the export ZIP file. Specify a value of \$True to scan for all the related objects and export them with the task sequence, including any dependencies for applications. To export only the task sequence XML without the other referenced objects, set this parameter to \$False.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Export a task sequence to a file

This command exports a sequence of tasks to a file and specifies an export file path.

```
PS C:\> Export-CMTaskSequence -Name "Task Sequence 1210" -ExportFilePath "\\Contoso-01\Users\AimeeLott\Desktop\TaskSequence.zip"
```

Related topics

[Disable-CMTaskSequence](#)

[Enable-CMTaskSequence](#)

[Get-CMTaskSequence](#)

[Import-CMTaskSequence](#)

[Remove-CMTaskSequence](#)

[Set-CMTaskSequence](#)

Export-CMUserCollection

Export-CMUserCollection

Exports a user collection in Configuration Manager.

Syntax

Parameter Set: SearchByNameMandatory

```
Export-CMUserCollection -ExportFilePath <String> -Name <String> [-ExportComment <String> ]  
[-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Export-CMUserCollection -CollectionId <String> -ExportFilePath <String> [-ExportComment  
<String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Export-CMUserCollection -ExportFilePath <String> -InputObject <IResultObject> [-  
ExportComment <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Export-CMUserCollection** cmdlet exports a user collection in Microsoft System Center 2012 Configuration Manager. To export a collection, you must have Read rights on the collection.

Parameters

-CollectionId<String>

Specifies an ID of a collection in Configuration Manager.

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

-ExportComment<String>

Specifies a comment for the export file.

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

-ExportFilePath<String>

Specifies a file path for the export file.

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

-InputObject<IResultObject>

Specifies an input object. To obtain a user collection object, use the [Get-CMUserCollection](#) cmdlet.

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

-Name<String>

Specifies a name for the export file.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Export a user collection

This command exports a user collection by using an export file path and a name.

```
PS C:\> Export-CMUserCollection -ExportFilePath "C:\Exports" -Name "All Western Users"
```

Example 2: Export a user collection by collection ID

This command exports a user collection by using a collection ID and an export file path.

```
PS C:\> Export-CMUserCollection -CollectionId "CM4" -ExportFilePath "C:\Exports"
```

Example 3: Export a user collection from an input object

The first command uses the **Get-CMUserCollection** to obtain an input object, and then stores it in the \$IRObj-013 variable.

The second command exports the user collection using the input object stored in \$IRObj-013.

```
PS C:\> $IRObj-013 = Get-CMUserCollection
```

```
PS C:\> Export-CMUserCollection -ExportFilePath "D:\Exported Collections" -InputObject $IRObj-013
```

Related topics

[Get-CMUserCollection](#)

[Import-CMUserCollection](#)

[New-CMUserCollection](#)

[Remove-CMUserCollection](#)

[Set-CMUserCollection](#)

Get-CMAccessAccount

Get-CMAccessAccount

Gets an access account.

Syntax

Parameter Set: SearchByApplicationName

```
Get-CMAccessAccount -ApplicationName <String> [-UserName <String> ] [ <CommonParameters>]
```

Parameter Set: SearchByApplication

```
Get-CMAccessAccount -Application <IResultObject> [-UserName <String> ] [ <CommonParameters>]
```

Parameter Set: SearchByApplicationId

```
Get-CMAccessAccount -ApplicationId <String> [-UserName <String> ] [ <CommonParameters>]
```

Parameter Set: SearchByBootImage

```
Get-CMAccessAccount -BootImage <IResultObject> [-UserName <String> ] [ <CommonParameters>]
```

Parameter Set: SearchByBootImageId

```
Get-CMAccessAccount -BootImageId <String> [-UserName <String> ] [ <CommonParameters>]
```

Parameter Set: SearchByBootImageName

```
Get-CMAccessAccount -BootImageName <String> [-UserName <String> ] [ <CommonParameters>]
```

Parameter Set: SearchByDriverPackage

```
Get-CMAccessAccount -DriverPackage <IResultObject> [-UserName <String> ] [ <CommonParameters>]
```

Parameter Set: SearchByDriverPackageId

```
Get-CMAccessAccount -DriverPackageId <String> [-UserName <String> ] [ <CommonParameters>]
```

Parameter Set: SearchByDriverPackageName

```
Get-CMAccessAccount -DriverPackageName <String> [-UserName <String> ] [ <CommonParameters>]
```

Parameter Set: SearchByOSImage

```
Get-CMAccessAccount -OperatingSystemImage <IResultObject> [-UserName <String> ] [ <CommonParameters>]
```

Parameter Set: SearchByOSImageId

```
Get-CMAccessAccount -OperatingSystemImageId <String> [-UserName <String> ] [ <CommonParameters>]
```

Parameter Set: SearchByOSImageName

```
Get-CMAccessAccount -OperatingSystemImageName <String> [-UserName <String> ] [ <CommonParameters>]
```

Parameter Set: SearchByOSInstaller

```
Get-CMAccessAccount -OperatingSystemInstaller <IResultObject> [-UserName <String> ] [ <CommonParameters>]
```

Parameter Set: SearchByOSInstallerId

```
Get-CMAccessAccount -OperatingSystemInstallerId <String> [-UserName <String> ] [ <CommonParameters>]
```

Parameter Set: SearchByOSInstallerName

```
Get-CMAccessAccount -OperatingSystemInstallerName <String> [-UserName <String> ] [ <CommonParameters>]
```

Parameter Set: SearchByPackage

```
Get-CMAccessAccount -Package <IResultObject> [-UserName <String> ] [ <CommonParameters>]
```

Parameter Set: SearchByPackageId

```
Get-CMAccessAccount -PackageId <String> [-UserName <String> ] [ <CommonParameters>]
```

Parameter Set: SearchByPackageName

```
Get-CMAccessAccount -PackageName <String> [-UserName <String> ] [ <CommonParameters>]
```

Parameter Set: SearchBySoftwareUpdateDeploymentPackage

```
Get-CMAccessAccount -SoftwareUpdateDeploymentPackage <IResultObject> [-UserName <String> ] [ <CommonParameters>]
```

Parameter Set: SearchBySoftwareUpdateDeploymentPackageId

```
Get-CMAccessAccount -SoftwareUpdateDeploymentPackageId <String> [-UserName <String> ] [ <CommonParameters>]
```

Parameter Set: SearchBySoftwareUpdateDeploymentPackageName

```
Get-CMAccessAccount -SoftwareUpdateDeploymentPackageName <String> [-UserName <String> ] [ <CommonParameters>]
```

Detailed Description

The **Get-CMAccessAccount** cmdlet gets an access account.

An access account is a list of users or groups that can access an established service or application that is located on a distribution point. For example, members in the Software Update Point Connection access account can access two services to manage software updates: Windows Server Update Services (WSUS) and WSUS Synchronization Manager. You can get an access account to change the network access permissions for clients who use the account.

Parameters

-Application<IResultObject>

Specifies a deployed application object. You can get an application object by using the **Get-CMAApplication** cmdlet.

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

-ApplicationId<String>

Specifies the ID of an application.

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

-ApplicationName<String>

Specifies the name of an application object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

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

-BootImage<IResultObject>

Specifies a boot image object. A boot image object contains the Windows files that are required to prepare a computer for the installation of an operating system. You can get a boot image object by using the **Get-CMBootImage** cmdlet.

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

-BootImageId<String>

Specifies the ID of a boot image object.

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

-BootImageName<String>

Specifies the name of a boot image object.

Aliases	none
Required?	true
Position?	named

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

-DriverPackage<IResultObject>

Specifies a driver package object. A driver package object specifies a group of hardware drivers that are required to install an operating system. You can get a driver package object by using the **Get-CMDriverPackage** cmdlet.

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

-DriverPackageld<String>

Specifies the ID of a driver package.

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

-DriverPackageName<String>

Specifies the name of a driver package.

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

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

-OperatingSystemImage<IResultObject>

Specifies an operating system image object. An operating system image object contains the Windows files that compose a complete Windows installation. You can get an operating system image object by using the **Get-CMOperatingSystemImage** cmdlet.

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

-OperatingSystemImageId<String>

Specifies the ID of an operating system image.

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

-OperatingSystemImageName<String>

Specifies the name of an operating system image.

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

-OperatingSystemInstaller<IResultObject>

Specifies an operating system installer object. An operating system installer object contains the Windows files that are required to prepare a computer for the installation of an operating system. To obtain an operating system installer object, use the **Get-CMOperatingSystemInstaller** cmdlet.

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

-OperatingSystemInstallerId<String>

Specifies the ID of an operating system installer.

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

-OperatingSystemInstallerName<String>

Specifies the name of an operating system installer object.

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

-Package<IResultObject>

Specifies a deployed software script or program object. You can get a package by using the **Get-CMPackage** cmdlet.

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

-Packageld<String>

Specifies the ID of a deployed software script or program object.

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

-PackageName<String>

Specifies the name of a deployed software script or program object.

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

-SoftwareUpdateDeploymentPackage<IResultObject>

Specifies a deployed software update object. You can get a software update object by using the **Get-CMSoftwareUpdate** cmdlet.

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

-SoftwareUpdateDeploymentPackageId<String>

Specifies the ID of a software update deployment object.

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

-SoftwareUpdateDeploymentPackageName<String>

Specifies the name of a deployed software update object.

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

-UserName<String>

Specifies a Windows user account name in *domain\user* format.

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

<CommonParameters>

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

Examples

Example 1: Get an access account for a package by using the package ID

In this example, the first command gets the package that is identified by using its name. The command stores the ID of the specified package in the \$ID variable.

The second command gets the access account for the identified package. The command output describes all users and groups that can access this package.

```
PS C:\> $ID = Get-CMPackage -PackageName "Configuration Manager Client Package"
PS C:\> Get-CMAccessAccount -PackageId $ID
```

```
Name: Administrator
Type: Administrator
Access: FullControl
Sitecode: CM1
PackageID: CM100002
Name: CONTOSO\PFuller
Type: WindowsUser
Access: Read
Sitecode: CM1
PackageID: CM100002
```

Example 2: Get an access account for a boot image by using the boot image name

The command gets the access account for a system boot image that is specified by using its name.

```
PS C:\> Get-CMAccessAccount -BootImageName "System Image (x64)"
```

```
Name: CONTOSO\EDaugherty
Type: WindowsUser
Access: Read
Sitecode: CM1
Boot Image: System Image (x64)
```

Related topics

[New-CMAccessAccount](#)

[Set-CMAccessAccount](#)

[Remove-CMAccessAccount](#)

[Get-CMApplication](#)

[Get-CMBootImage](#)

[Get-CMDriverPackage](#)

[Get-CMOperatingSystemImage](#)

[Get-CMOperatingSystemInstaller](#)

[Get-CMPackage](#)

[Get-CMSoftwareUpdate](#)

[Get-CMSoftwareUpdateDeploymentPackage](#)

Get-CMAccessLicense

Get-CMAccessLicense

Gets license usage information.

Syntax

Parameter Set: ByValue

```
Get-CMAccessLicense -LicenseName <String> [ <CommonParameters>]
```

Parameter Set: ByCount

```
Get-CMAccessLicense -Count -LicenseName <String> [ <CommonParameters>]
```

Parameter Set: ByName

```
Get-CMAccessLicense -License [ <CommonParameters>]
```

Detailed Description

The **Get-CMAccessLicense** cmdlet gets license usage information for the servers and clients in the scope of System Center 2012 Configuration Manager. **Get-CMAccessLicense** returns a list of features able to be licensed and a list of unique users and devices per licensable feature.

Parameters

-Count

Indicates that the cmdlet returns a count of unique users and devices for the specified licensable feature.

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

-License

Indicates that the cmdlet gets all licensable features for all of the servers and clients within the scope of System Center 2012 Configuration Manager. You can pass the name of the license that is returned to the *LicenseName* parameter to obtain the unique users and devices for that specific license.

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

-LicenseName<String>

Specifies the name of a licensable feature. If specified, the cmdlet gets only the unique users and devices for the specified license name. Valid values are:

- ConfigMgr_2012_CoreServer
- ConfigMgr_2012_CoreClient
- ConfigMgr_2012_EndpointClient

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

Example 1: Get all licensable features for all servers and clients

This command gets all licensable features for all servers and clients within the scope of System Center 2012 Configuration Manager.

```
PS C:\> Get-CMAccessLicense -License
```

Example 2: Get the unique users, devices, and license-specific unique ID for a specified license

This command gets the unique users, devices, and license-specific IDs for the license named ConfigMgr_2012_EndPointClient.

```
PS C:\> Get-CMAccessLicense -LicenseName ConfigMgr_2012_EndPointClient
```

Get-CMAccount

Get-CMAccount

Gets a named user account.

Syntax

Parameter Set: SearchByName

```
Get-CMAccount [-Name <String> ] [ <CommonParameters>]
```

Detailed Description

The **Get-CMAccount** cmdlet gets a Microsoft System Center 2012 Configuration Manager user account. The user name must be in the *domain\user* format. For more information about System Center 2012 Configuration Manager user accounts, see [Technical Reference for Accounts Used in Configuration Manager](http://go.microsoft.com/fwlink/?LinkID=248317) (<http://go.microsoft.com/fwlink/?LinkID=248317>) on TechNet.

Parameters

-Name<String>

Specifies the name of the user account.

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

<CommonParameters>

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

Examples

Example 1: Get a user account by using its name

This command gets a user account.

```
PS C:\> Get-CMAccount -Name "CONTOSO\ENarvaez"
```

Related topics

[New-CMAccount](#)

[Remove-CMAccount](#)

[Set-CMAccount](#)

Get-CMActiveDirectoryForest

Get-CMActiveDirectoryForest

Gets one or more Active Directory forest objects.

Syntax

Parameter Set: SearchByFQDN

```
Get-CMActiveDirectoryForest [-ForestFqdn <String> ] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Get-CMActiveDirectoryForest -Id <String[]> [ <CommonParameters>]
```

Detailed Description

The **Get-CMActiveDirectoryForest** cmdlet gets one or more Active Directory forest objects in Microsoft System Center 2012 Configuration Manager. You can get an Active Directory forest object by ID or fully qualified domain name (FQDN).

Parameters

-ForestFqdn<String>

Specifies the FQDN of a Configuration Manager object.

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

-Id<String[]>

Specifies an array of IDs of Configuration Manager objects. You can find the identifier value in the ForestID property of an Active Directory forest.

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

<CommonParameters>

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

Examples

Example 1: Get all Active Directory forest objects

This command gets all Active Directory forest objects.

```
PS C:\> Get-CMActiveDirectoryForest
```

Example 2: Get an Active Directory Forest object by ID

This command gets an Active Directory forest object that has the ID 16777217.

```
PS C:\> Get-CMActiveDirectoryForest -Id "16777217"
```

Example 3: Get Active Directory Forest by domain name

This command gets an Active Directory forest object that has the FQDN tsqa.contoso.com.

```
PS C:\> Get-CMActiveDirectoryForest -ForestFqdn "tsqa.contoso.com"
```

Related topics

[New-CMActiveDirectoryForest](#)

[Set-CMActiveDirectoryForest](#)

[Remove-CAActiveDirectoryForest](#)

Get-CMActiveDirectorySite

Get-CMActiveDirectorySite

Gets Configuration Manager sites that publish data to AD DS.

Syntax

Parameter Set: SearchByName

```
Get-CMActiveDirectorySite [-Name <String[]> ] [ <CommonParameters>]
```

Parameter Set: SearchByForestId

```
Get-CMActiveDirectorySite -ForestId <String[]> [ <CommonParameters>]
```

Parameter Set: SearchByForestName

```
Get-CMActiveDirectorySite -ForestFqdn <String[]> [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Get-CMActiveDirectorySite -Id <String[]> [ <CommonParameters>]
```

Detailed Description

The **Get-CMActiveDirectorySite** cmdlet gets one or more Microsoft System Center 2012 Configuration Manager sites that are configured to publish site information to Active Directory Domain Services (AD DS). You can get System Center 2012 Configuration Manager sites that publish site data to AD DS by using an identifier or a fully qualified domain name (FQDN).

Parameters

-ForestFqdn<String[]>

Specifies an array of fully qualified domain names that identify AD forests. The FQDN provides a path to an AD forest.

Aliases	none
Required?	true
Position?	named
Default Value	none

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

-ForestId<String[]>

Specifies an array of IDs that identify AD forests.

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

-Id<String[]>

Specifies an array of identifiers of Active Directory forest objects that contain Active Directory sites.

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

-Name<String[]>

Specifies an array of FQDNs of Active Directory forest objects that contain Active Directory sites.

Aliases	ADSiteName
Required?	false
Position?	named
Default Value	none

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

<CommonParameters>

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

Examples

Example 1: Get an Active Directory site

This command gets the Active Directory sites that are configured to publish site information.

```
PS C:\> Get-CMActiveDirectorySite
```

Related topics

[Get-CMActiveDirectoryForest](#)

Get-CMAdministrativeUser

Get-CMAdministrativeUser

Gets Configuration Manager administrative users.

Syntax

Parameter Set: SearchByName

```
Get-CMAdministrativeUser [-Name <String[]> ] [-RoleName <String[]> ] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Get-CMAdministrativeUser -Id <String[]> [-RoleName <String[]> ] [ <CommonParameters>]
```

Detailed Description

The **Get-CMAdministrativeUser** cmdlet gets one or more administrative users in Microsoft System Center 2012 Configuration Manager. An administrative user has a defined set of permissions and may be a member of one or more scopes or roles.

Parameters

-Id<String[]>

Specifies an array of IDs for administrative users.

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

-Name<String[]>

Specifies an array of administrative user names in the form of *<domain>\<user>*.

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

-RoleName<String[]>

Specifies an array of names of security roles. Valid values are:

- Application Administrator
- Application Author
- Application Deployment Manager
- Asset Manager
- Compliance Settings Manager
- Discovery Operator
- Endpoint Protection Manager
- Full Administrator
- Infrastructure Administrator
- Operating System Deployment Manager
- Operations Administrator
- Read-only Analyst
- Remote Tools Operator
- Security Administrator
- Software Update Manager
- custom-defined security roles

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

<CommonParameters>

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

Examples

Example 1: Get an administrative user by using an ID

This command gets the administrative user who has the ID 16777217 and who is assigned to the Application Author security role.

```
PS C:\> Get-CMAdministrativeUser -Id "16777217" -RoleNames "Application Author"
```

Example 2: Get an administrative user by using a wildcard

This command gets administrative users who have a display name that begins with the letter D.

```
PS C:\> Get-CMAdministrativeUser -Name D*
```

Example 3: Get an administrative user by using a security role assignment

This command gets administrative users who are assigned to the Application Administrator security role.

```
PS C:\> Get-CMAdministrativeUser -RoleNames "Application Administrator"
```

Related topics

[New-CMAdministrativeUser](#)

[Remove-CMAdministrativeUser](#)

Get-CMAAlert

Get-CMAAlert

Gets Configuration Manager alerts.

Syntax

Parameter Set: SearchByName

```
Get-CMAAlert [-Name <String> ] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Get-CMAAlert -Id <String> [ <CommonParameters>]
```

Detailed Description

The **Get-CMAAlert** cmdlet gets one or more Microsoft System Center 2012 Configuration Manager alerts. You can get a specific alert by specifying the name or identifier of the alert.

Parameters

-Id<String>

Specifies an alert identifier. You can obtain the identifier of an alert by using the **Get-Alert** cmdlet.

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

-Name<String>

Specifies an alert name. You can obtain the name of an alert by using **Get-CMAAlert**.

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

<CommonParameters>

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

Examples

Example 1: Get all alerts

This command gets all alerts that System Center 2012 Configuration Manager manages.

```
PS C:\> Get-CMAAlert
```

Example 2: Get alerts by using name

This command gets all alerts that have a name that begins with the letter D.

```
PS C:\> Get-CMAAlert -Name D*
```

Related topics

[Enable-CMAAlert](#)

[Remove-CMAAlert](#)

[Set-CMAAlert](#)

[Suspend-CMAAlert](#)

[Disable-CMAAlert](#)

Get-CMAAlertSubscription

Get-CMAAlertSubscription

Gets one or more alert subscription objects.

Syntax

Parameter Set: SearchByName

```
Get-CMAAlertSubscription [-Name <String> ] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Get-CMAAlertSubscription -Id <String> [ <CommonParameters>]
```

Detailed Description

The **Get-CMAAlertSubscription** cmdlet gets one or more Microsoft System Center 2012 Configuration Manager alert subscriptions and displays their properties. If you specify the name or ID of an alert subscription, the cmdlet retrieves only that alert subscription. If you specify part of the name or ID of an alert subscription, the cmdlet retrieves all alert subscriptions that match the partial name or ID. If you do not specify anything, the cmdlet returns the properties of all alert subscriptions.

Parameters

-Id<String>

Specifies the identifier of a subscription.

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

-Name<String>

Specifies the name of an alert subscription object.

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

<CommonParameters>

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

Examples

Example 1: Display all alert subscriptions

This command displays all System Center 2012 Configuration Manager alert subscriptions.

```
PS C:\> Get-CMAAlertSubscription
```

Example 2: Display alert subscriptions by ID via wildcards

This command displays a list of all System Center 2012 Configuration Manager alert subscriptions that begin with the number 16777.

```
PS C:\> Get-CMAAlertSubscription -Id 16777*
```

Example 3: Display an alert subscription by name

This command displays the System Center 2012 Configuration Manager alert subscription named Subscription01.

```
PS C:\> Get-CMAAlertSubscription -Name "Subscription01"
```

Related topics

[New-CMAAlertSubscription](#)

[Set-CMAAlertSubscription](#)

[Remove-CMAAlertSubscription](#)

Get-CMAntiMalwarePolicy

Get-CMAntiMalwarePolicy

Gets antimalware policies for Endpoint Protection.

Syntax

Parameter Set: SearchByName

```
Get-CMAntiMalwarePolicy [-Name <String> ] [-Policy {Advanced | DefaultActions | DefinitionUpdates | ExclusionSettings | MicrosoftActiveProtectionService | RealTimeProtection | ScanSettings | ScheduledScans | ThreatOverrides} ] [-SecuredScopeNames <String> ] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Get-CMAntiMalwarePolicy -Id <String> [-Policy {Advanced | DefaultActions | DefinitionUpdates | ExclusionSettings | MicrosoftActiveProtectionService | RealTimeProtection | ScanSettings | ScheduledScans | ThreatOverrides} ] [-SecuredScopeNames <String> ] [ <CommonParameters>]
```

Detailed Description

The **Get-CMAntiMalwarePolicy** cmdlet gets the antimalware policies for System Center 2012 Endpoint Protection that apply to collections of client computers that are running the Microsoft System Center 2012 Configuration Manager agent. Antimalware policies are configuration settings that define how an antimalware agent operates on client computers.

Parameters

-Id<String>

Specifies the identity of an antimalware policy object.

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

-Name<String>

Specifies the name of an antimalware policy.

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

-Policy<PolicyType[]>

Specifies an array of policy settings for antimalware. Valid values are:

- Advanced
- DefaultActions
- DefinitionUpdates
- ExclusionSettings
- MicrosoftActiveProtectionService
- RealTimeProtection
- ScanSettings
- ThreatOverrides

The acceptable values for this parameter are:

Advanced	
DefaultActions	
DefinitionUpdates	
ExclusionSettings	
MicrosoftActiveProtectionService	
RealTimeProtection	
ScanSettings	
ScheduledScans	
ThreatOverrides	

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

-SecuredScopeNames<String>

Specifies the names of secured scopes.

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

<CommonParameters>

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

Examples

Example 1: Get all antimalware policies

This command gets all antimalware policies for Endpoint Protection in Configuration Manager.

```
PS C:\> Get-CMAntiMalwarePolicy
```

Example 2: Get antimalware policies by using a wildcard

This command gets antimalware policies that have a name that begins with the letter D.

```
PS C:\> Get-CMAntiMalwarePolicy -Name D*
```

Example 3: Get antimalware policies by using an object ID

This command gets the antimalware policy that has the ID 16777217.

```
PS C:\> Get-CMAntiMalwarePolicy -Id "16777217"
```

Related topics

[New-CMAntimalwarePolicy](#)

[Set-CMAntiMalwarePolicy](#)

[Export-CMAntimalwarePolicy](#)

[Import-CMAntimalwarePolicy](#)

[Merge-CMAntimalwarePolicy](#)

[Remove-CMAntiMalwarePolicy](#)

Get-CMApplication

Get-CMApplication

Gets properties of an application in Configuration Manager.

Syntax

Parameter Set: SearchByName

```
Get-CMApplication [-Name <String[]> ] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Get-CMApplication -Id <String[]> [ <CommonParameters>]
```

Detailed Description

The **Get-CMApplication** cmdlet gets properties of an application.

Parameters

-Id<String[]>

Specifies an array of application IDs.

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

-Name<String[]>

Specifies an array of application names.

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

<CommonParameters>

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

Examples

Example 1: Retrieve all applications

This command retrieves all System Center 2012 Configuration Manager applications.

```
PS C:\> Get-CMApplication
```

Related topics

[Export-CMApplication](#)

[Import-CMApplication](#)

[New-CMApplication](#)

[Remove-CMApplication](#)

[Resume-CMApplication](#)

[Set-CMApplication](#)

[Suspend-CMApplication](#)

Get-CMApplicationCatalogWebServicePoint

Get-CMApplicationCatalogWebServicePoint

Gets an Application Catalog web service point.

Syntax

Parameter Set: SearchByName

```
Get-CMApplicationCatalogWebServicePoint [-SiteCode <String> ] [-SiteSystemServerName <String> ] [ <CommonParameters>]
```

Detailed Description

The **Get-CMApplicationCatalogWebServicePoint** cmdlet gets a Microsoft System Center 2012 Configuration Manager Application Catalog web service point object that has a specified site code for a fully qualified domain name (FQDN).

Before you can configure an Application Catalog web service point you must first install and configure site system roles in System Center 2012 Configuration Manager. For more information, see [Install and Configure Site System Roles for Configuration Manager](http://go.microsoft.com/fwlink/?LinkId=262649) (<http://go.microsoft.com/fwlink/?LinkId=262649>) on TechNet.

Parameters

-SiteCode<String>

Specifies a site code for an Application Catalog web service point object.

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

-SiteSystemServerName<String>

Specifies an FQDN for an Application Catalog web service point.

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

<CommonParameters>

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

Examples

Example 1: Get a system role

This command gets an Application Catalog web service point named western.contoso.com that has the site code CM1.

```
PS C:\> Get-CMApplicationCatalogWebServicePoint -SiteSystemServerName "western.contoso.com" -SiteCode "CM1"
```

Related topics

[Add-CMApplicationCatalogWebServicePoint](#)

[Remove-CMApplicationCatalogWebServicePoint](#)

Get-CMApplicationCatalogWebsitePoint

Get-CMApplicationCatalogWebsitePoint

Gets a Configuration Manager Application Catalog website point.

Syntax

Parameter Set: SearchByName

```
Get-CMApplicationCatalogWebsitePoint [-SiteCode <String> ] [-SiteSystemServerName <String> ]  
[ <CommonParameters>]
```

Detailed Description

The **Get-CMApplicationCatalogWebsitePoint** cmdlet gets an Application Catalog website point in Microsoft System Center 2012 Configuration Manager. This site system role supports the Application Catalog website.

You can specify a website point by either site code or by the name of the server that hosts the role. Use this cmdlet with no parameters to get all Application Catalog website points for a System Center 2012 Configuration Manager hierarchy.

Parameters

-SiteCode<String>

Specifies the site code for a Configuration Manager site.

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

-SiteSystemServerName<String>

Specifies the name of a server that hosts a site system role.

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

<CommonParameters>

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

Examples

Example 1: Get a website point by using a site code

This command gets the website point role for the site that has the site code CM4.

```
PS C:\> Get-CMApplicationCatalogWebsitePoint -SiteCode "CM4"
```

Example 2: Get a website point by using a site system name

This command gets the website point role that the computer WesternACWP.Contoso.com hosts.

```
PS C:\> Get-CMApplicationCatalogWebsitePoint -SiteSystemServerName "WesternACWP.Contoso.com"
```

Related topics

[Add-CMApplicationCatalogWebsitePoint](#)

[Remove-CMApplicationCatalogWebSitePoint](#)

[Set-CMApplicationCatalogWebsitePoint](#)

[Add-CMApplicationCatalogWebServicePoint](#)

[Get-CMApplicationCatalogWebServicePoint](#)

Get-CMApplicationRevisionHistory

Get-CMApplicationRevisionHistory

Gets a Configuration Manager object that represents the revision history for an application.

Syntax

Parameter Set: SearchByNameMandatory

```
Get-CMApplicationRevisionHistory -Name <String[]> [-Revision <Int32> ] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Get-CMApplicationRevisionHistory -Id <String[]> [-Revision <Int32> ] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Get-CMApplicationRevisionHistory -InputObject <IResultObject> [-Revision <Int32> ] [ <CommonParameters>]
```

Detailed Description

The **Get-CMApplicationRevisionHistory** cmdlet gets a Microsoft System Center 2012 Configuration Manager object that represents the revision history for an application. When you revise an application or a deployment type contained in an application, System Center 2012 Configuration Manager creates a new revision of the application. You can use the revision history to display each revision made to an application, view the properties of a revision, restore a previous revision, or delete an old revision.

Parameters

-Id<String[]>

Specifies an array of IDs of application revision histories.

Aliases	CId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

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

-InputObject<IResultObject>

Specifies an application object. To obtain an application object, use the **Get-CMApplication** cmdlet.

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

-Name<String[]>

Specifies an array of names of application revision histories.

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

-Revision<Int32>

Specifies the version number of an application revision.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?

false

<CommonParameters>

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

Examples

Example 1: Get the revision history for an application

This command gets the application revision history named MSXML 6.0 Parser.

```
PS C:\> Get-CMApplicationRevisionHistory -Name "MSXML 6.0 Parser"
```

Related topics

[Remove-CMApplicationRevisionHistory](#)

[Restore-CMApplicationRevisionHistory](#)

Get-CMApprovalRequest

Get-CMApprovalRequest

Gets a request to allow the installation of an application.

Syntax

Parameter Set: SearchByName

```
Get-CMApprovalRequest [-ApplicationName <String[]> ] [-User <String> ] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Get-CMApprovalRequest -Id <String[]> [ <CommonParameters>]
```

Detailed Description

The **Get-CMApprovalRequest** cmdlet gets a request from a user to install an application. You can specify an approval request by application name, application ID, or by user name.

Parameters

-ApplicationName<String[]>

Specifies an array of names of applications.

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

-Id<String[]>

Specifies an array of IDs of applications.

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

-User<String>

Specifies an array of user names of persons who submitted the approval request. Use the format *domain\user*.

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

<CommonParameters>

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

Examples

Example 1: Get all approval requests

This command gets all pending Microsoft System Center 2012 Configuration Manager approval requests.

```
PS C:\> Get-CMApprovalRequest
```

Example 2: Get an approval request by using an application ID

This command gets an approval request for an application with the specified ID.

```
PS C:\> Get-CMApprovalRequest -Id "1635223"
```

Example 3: Get an approval request for a specific user

This command gets an approval request for the application HelloWorld for a specified user.

```
PS C:\> Get-CMApprovalRequest -Application "HelloWorld" -User "tsqa\davidchew"
```

Related topics

[Approve-CMApprovalRequest](#)

[Deny-CMApprovalRequest](#)

Get-CMAppV5XDeploymentTypeItem

Get-CMAppV5XDeploymentTypeItem

Gets application information and deployment types from an App-V 5.0 package file.

Syntax

Parameter Set: GetByApplicationName

```
Get-CMAppV5XDeploymentTypeItem -ApplicationName <String> -DeploymentTypeName <String> [  
<CommonParameters>]
```

Detailed Description

The **Get-CMAppV5XDeploymentTypeItem** cmdlet gets application information and deployment types from a Microsoft Application Virtualization (App-V) 5.0 package file.

Parameters

-ApplicationName<String>

Specifies the name of an application that is associated to the App-V 5.0 deployment type.

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

-DeploymentTypeName<String>

Specifies the name of a deployment type.

A deployment type is contained within an application and contains the information that Configuration Manager requires to install software. A deployment type also contains rules that specify if and how the software is deployed.

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

<CommonParameters>

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

Examples

Example 1: Get deployment types from an App-V 5.0 package file

This command gets information for the application named BasicOfficeApps that is associated with deployment type named VirtualAppDeployment01.

```
PS C:\> Get-CMAppV5XDeploymentTypeItem -ApplicationName "BasicOfficeApps" -  
DeploymentTypeName "VirtualAppDeployment01"
```

Related topics

[Add-CMDeploymentType](#)

Get-CMAppVVirtualEnvironment

Get-CMAppVVirtualEnvironment

Gets an App-V virtual environment.

Syntax

Parameter Set: SearchByName

```
Get-CMAppVVirtualEnvironment [-Name <String[]> ] [-SecuredScopeNames <String> ] [
<CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Get-CMAppVVirtualEnvironment -Id <String[]> [-SecuredScopeNames <String> ] [
<CommonParameters>]
```

Detailed Description

The **Get-CMAppVVirtualEnvironment** cmdlet gets one or more Microsoft Application Virtualization (App-V) virtual environment objects from Microsoft System Center 2012 Configuration Manager. You can specify App-V environments by name or ID.

Parameters

-Id<String[]>

Specifies an array of IDs of virtual environments.

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

-Name<String[]>

Specifies an array of names of App-V virtual environment objects. You can use a wildcard.

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

-SecuredScopeNames<String>

Specifies the name of security scopes. A security scope can be Default or the name of a custom-created security scope.

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

<CommonParameters>

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

Examples

Example 1: Get all virtual environments

This command gets all App-V virtual environments.

```
PS C:\> Get-CMAppVVirtualEnvironment
```

Example 2: Get virtual environments by using a wildcard

This command gets all App-V virtual environments that have names that begin with the letter T.

```
PS C:\> Get-CMAppVVirtualEnvironment -Name "T*"
```

Example 3: Get virtual environment by an ID

This command gets an App-V virtual environment that has the ID 16781806.

```
PS C:\> Get-CMAppVVirtualEnvironment -Id "16781806"
```

Related topics

[New-CMAppVVirtualEnvironment](#)

[Remove-CMAppVVirtualEnvironment](#)

[Set-CMAppVVirtualEnvironment](#)

Get-CMAssetIntelligenceCatalogItem

Get-CMAssetIntelligenceCatalogItem

Gets an item from the Asset Intelligence catalog.

Syntax

Parameter Set: SearchByName

```
Get-CMAssetIntelligenceCatalogItem [-CategoryName <String> ] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Get-CMAssetIntelligenceCatalogItem -Id <String[]> [ <CommonParameters>]
```

Detailed Description

The **Get-CMAssetIntelligenceCatalogItem** cmdlet gets software categories, software families, and custom software labels from the Asset Intelligence catalog in Microsoft System Center 2012 Configuration Manager.

The Asset Intelligence catalog contains categorization and identification information for software titles. The catalog includes predefined categories and families. Predefined items cannot be modified. In addition to predefined software categories and software families, you can create custom categories and families. You can also create custom software labels.

For more information about the Asset Intelligence catalog, see [Introduction to Asset Intelligence in Configuration Manager](http://go.microsoft.com/fwlink/?LinkId=262650) (<http://go.microsoft.com/fwlink/?LinkId=262650>) on TechNet.

Parameters

-CategoryName<String>

Specifies the name of a category, family, or label in the Asset Intelligence catalog.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

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

-Id<String[]>

Specifies an array of IDs of asset intelligence catalog items.

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

<CommonParameters>

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

Examples

Example 1: Get catalog items by category name

This command gets Asset Intelligence catalog items by category name.

```
PS C:\> Get-CMAssetIntelligenceCatalogItem -CategoryName "Browsers"
```

Example 2: Get catalog items by category ID

This command gets Asset Intelligence catalog items by category ID.

```
PS C:\> Get-CMAssetIntelligenceCatalogItem -Id "1211"
```

Related topics

[New-CMAssetIntelligenceCatalogItem](#)

[Set-CMAssetIntelligenceCatalogItem](#)

[Remove-CMAssetIntelligenceCatalogItem](#)

Get-CMAssetIntelligenceSynchronizationPoint

Get-CMAssetIntelligenceSynchronizationPoint

Gets Asset Intelligence synchronization points.

Syntax

```
Get-CMAssetIntelligenceSynchronizationPoint [ <CommonParameters>]
```

Detailed Description

The **Get-CMAssetIntelligenceSynchronizationPoint** cmdlet gets one or more Asset Intelligence synchronization points. Microsoft System Center 2012 Configuration Manager uses the Asset Intelligence synchronization point site system role to connect System Center 2012 Configuration Manager sites to System Center Online to synchronize Asset Intelligence catalog information.

Parameters

<CommonParameters>

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

Examples

Example 1: Get an Asset Intelligence synchronization point

This command gets an Asset Intelligence synchronization point.

```
PS C:\> Get-CMAssetIntelligenceSynchronizationPoint
```

Related topics

[Set-CMAssetIntelligenceSynchronizationPoint](#)

[Add-CMAssetIntelligenceSynchronizationPoint](#)

[Remove-CMAssetIntelligenceSynchronizationPoint](#)



Get-CMAutomaticAmtProvisioningStatus

Get-CMAutomaticAmtProvisioningStatus

Gets the automatic provisioning status of computers with an AMT management controller.

Syntax

Parameter Set: SearchByNameMandatory

```
Get-CMAutomaticAmtProvisioningStatus -DeviceName <String[]> [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Get-CMAutomaticAmtProvisioningStatus -DeviceId <String[]> [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Get-CMAutomaticAmtProvisioningStatus -InputObject <IResultObject> [ <CommonParameters>]
```

Detailed Description

The **Get-CMAutomaticAmtProvisioningStatus** cmdlet gets the current automatic provisioning status of one or more computers with an AMT management controller.

Parameters

-DeviceId<String[]>

Specifies an array of device IDs.

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

-DeviceName<String[]>

Specifies an array of device names.

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

-InputObject<IResultObject>

Specifies a device object. To obtain a **CMDevice** object, use the [Get-CMDevice](#) cmdlet.

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

<CommonParameters>

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

Examples

Example 1: Get the automatic provisioning status of a computer

This command gets the automatic provisioning status of a computer with an AMT management controller named CMDIV-WEST03.

```
PS C:\> Get-CMAutomaticAmtProvisioningStatus -DeviceName "CMDIV-WEST03"
```

Related topics

[Enable-CMAutomaticAMTProvisioning](#)

[Get-CMDevice](#)

Get-CMBaseline

Get-CMBaseline

Gets configuration baselines.

Syntax

Parameter Set: SearchByName

```
Get-CMBaseline [-Name <String[]> ] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Get-CMBaseline -Id <String[]> [ <CommonParameters>]
```

Parameter Set: SearchByParentBaseline

```
Get-CMBaseline -ParentBaseline <IResultObject> [ <CommonParameters>]
```

Parameter Set: SearchByParentBaselineIdMandatory

```
Get-CMBaseline -ParentBaselineId <String> [ <CommonParameters>]
```

Parameter Set: SearchByParentBaselineNameMandatory

```
Get-CMBaseline -ParentBaselineName <String> [ <CommonParameters>]
```

Detailed Description

The **Get-CMBaseline** cmdlet gets one or more configuration baselines.

Parameters

-Id<String[]>

Specifies an array of IDs of configuration baselines.

Aliases	CllId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

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

-Name<String[]>

Specifies an array of names of configuration baselines.

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

-ParentBaseline<IResultObject>

Specifies a **CMParentBaseline** object.

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

-ParentBaselineId<String>

Specifies the ID of a parent baseline object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

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

-ParentBaselineName<String>

Specifies the name of a parent baseline.

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

<CommonParameters>

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

Examples

Example 1: Get configuration baselines by using a parent baseline name

This command gets the child configuration baselines in the parent baseline configuration item named ParentBaselineContoso01.

```
PS C:\> Get-CMBaseline -ParentBaselineName "ParentBaselineContoso01"
```

Example 2: Get configuration baselines by using a parent baseline ID

This command gets the child configuration baselines in the parent baseline configuration item that has the identity 16777357.

```
PS C:\> Get-CMBaseline -ParentBaselineId "16777357"
```

Related topics

[Disable-CMBaseline](#)

[Enable-CMBaseline](#)

[Export-CMBaseline](#)

[Import-CMBaseline](#)

[New-CMBaseline](#)

[Remove-CMBaseline](#)

[Set-CMBaseline](#)

[Get-CMBaselineSummarizationSchedule](#)

Get-CMBaselineSummarizationSchedule

Get-CMBaselineSummarizationSchedule

Gets the summarization schedule for configuration baseline data.

Syntax

```
Get-CMBaselineSummarizationSchedule [ <CommonParameters> ]
```

Detailed Description

The **Get-CMBaselineSummarizationSchedule** cmdlet gets the schedule by which the configuration baseline data in the Microsoft System Center 2012 Configuration Manager is updated with the latest information from the site database.

Parameters

<CommonParameters>

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

Examples

Example 1: Get the update schedule for configuration baseline data

This command gets the update schedule for configuration baseline data.

```
PS C:\> Get-CMBaselineSummarizationSchedule
```

Related topics

[Set-CMBaselineSummarizationSchedule](#)

[Invoke-CMBaselineSummarization](#)

Get-CMBaselineXMLDefinition

Get-CMBaselineXMLDefinition

Gets the XML definition of a configuration baseline.

Syntax

Parameter Set: SearchByIdMandatory

```
Get-CMBaselineXMLDefinition -Id <String[]> [ <CommonParameters>]
```

Parameter Set: SearchByName

```
Get-CMBaselineXMLDefinition [-Name <String[]> ] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Get-CMBaselineXMLDefinition -InputObject <IResultObject> [ <CommonParameters>]
```

Detailed Description

The **Get-CMBaselineXMLDefinition** cmdlet gets and displays the XML definition of one or more baseline configurations.

Parameters

-Id<String[]>

Specifies an array of IDs of baseline configurations.

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

-InputObject<IResultObject>

Specifies a **CMBaseline** object. To obtain a **CMBaseline** object, use the **Get-CMBaseline** cmdlet.

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

-Name<String[]>

Specifies an array of baseline configuration names.

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

<CommonParameters>

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

Examples

Example 1: Get a configuration baseline XML definition

The first command gets the configuration baseline object that has the ID 16777568, and stores the object in the \$CIObj variable.

The second command gets the XML definition of the configuration baseline stored in \$CIObj.

```
PS C:\> $CIObj = Get-CMBaseline -Id "16777568"  
PS C:\> Get-CMBaselineXMLDefinition -InputObject $CIObj
```

Related topics

[Get-CMBaseline](#)

Get-CMBootImage

Get-CMBootImage

Gets an operating system boot image.

Syntax

Parameter Set: SearchByName

```
Get-CMBootImage [-Name <String> ] [-SecuredScopeNames <String> ] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Get-CMBootImage -Id <String[]> [-SecuredScopeNames <String> ] [ <CommonParameters>]
```

Detailed Description

The **Get-CMBootImage** cmdlet gets a Windows PE operating system boot image that Microsoft System Center 2012 Configuration Manager can use to deploy an operating system.

Operating system boot images are .wim format files. These files contain a compressed set of reference files and folders that are required to successfully install and configure an operating system image. By default, System Center 2012 Configuration Manager includes both x86 and x64 boot images.

You must run the **Get-CMBootImage** cmdlet on the computer that is running the Systems Management Server (SMS) provider. The computer account of the computer that is running the SMS provider must have Read and Write access to the source package of the boot image. For more information about the SMS provider, see [Planning for the SMS Provider in Configuration Manager](http://go.microsoft.com/fwlink/?LinkID=263566) (<http://go.microsoft.com/fwlink/?LinkID=263566>) on TechNet.

Parameters

-Id<String[]>

Specifies an array of boot image identifiers.

Aliases	PackageId
Required?	true
Position?	named
Default Value	none

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

-Name<String>

Specifies a name of a boot image.

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

-SecuredScopeNames<String>

Specifies secured scope name.

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

<CommonParameters>

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

Examples

Example 1: Get a boot image by using its ID

This command gets a boot image by using its ID.

```
PS C:\> Get-CMBootImage -Id "c0eb2912-0de8-4a2a-9c77-603b35bcf7e4"
```

Example 2: Get a boot image by using its name

This command gets a boot image by using its name.

```
PS C:\> Get-CMBootImage -Name "SMS_BootImagePackage"
```

Related topics

[New-CMBootImage](#)

[Remove-CMBootImage](#)

[Set-CMBootImage](#)

Get-CMBoundary

Get-CMBoundary

Gets a boundary.

Syntax

Parameter Set: SearchByName

```
Get-CMBoundary [-Name <String[]> ] [ <CommonParameters>]
```

Parameter Set: SearchByBoundaryGroup

```
Get-CMBoundary -BoundaryGroup <IResultObject> [ <CommonParameters>]
```

Parameter Set: SearchByBoundaryGroupIdMandatory

```
Get-CMBoundary -BoundaryGroupId <String> [ <CommonParameters>]
```

Parameter Set: SearchByBoundaryGroupNameMandatory

```
Get-CMBoundary -BoundaryGroupName <String> [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Get-CMBoundary -Id <String[]> [ <CommonParameters>]
```

Detailed Description

The **Get-CMBoundary** cmdlet gets a boundary.

In Microsoft System Center 2012 Configuration Manager, a boundary is an intranet location that contains one or more devices that you can manage. A boundary can be an IP subnet, Active Directory site name, IPv6 prefix, or an IP address range.

Parameters

-BoundaryGroup<IResultObject>

Specifies an input object to this cmdlet, which in this case is a boundary group (a collection of boundaries). You can get a boundary group object by using the **Get-CMBoundaryGroup** cmdlet.

Aliases	none
Required?	true

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-BoundaryGroupId<String>

Specifies an identifier (ID) for a boundary group. You can get a boundary group ID by using the **Get-CMBoundaryGroup** cmdlet.

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

-BoundaryGroupName<String>

Specifies a name for a boundary group. You can get a boundary group name by using the **Get-CMBoundaryGroup** cmdlet.

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

-Id<String[]>

Specifies an array of boundary IDs.

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

-Name<String[]>

Specifies an array of boundary names.

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

<CommonParameters>

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

Examples

Example 1: Get a boundary that is specified by its identifier.

This command gets a boundary that is specified by the identifier 67777217.

```
PS C:\> Get-Boundary -Id "67777217"
```

```
BoundaryFlags:      0
BoundaryID:        67777217
BoundaryType:      1
CreatedBy:         Contoso\PFuller
CreatedOn:         6/10/2012 2:58:56 PM
DefaultSiteCode:
```

DisplayName:
GroupCount: 0
ModifiedBy: Contoso\PFuller
ModifiedOn: 9/13/2012 10:04 AM
SiteSystems:
Value: Default1

Example 2: Get a boundary that is specified by the name of an associated boundary group

This command gets a boundary that is specified by the associated boundary group BGroup07.

PS C:\> Get-Boundary -BoundaryGroupName "BGroup07"

BoundaryFlags: 0
BoundaryID: 63997411
BoundaryType: 2
CreatedBy: Contoso\PFuller
CreatedOn: 4/13/2012 06:58:56 AM
DefaultSiteCode:
DisplayName:
GroupCount: 1
ModifiedBy: Contoso\PFuller
ModifiedOn: 8/02/2012 11:16 AM
SiteSystems:
Value: Default1

Related topics

[Remove-CMBoundary](#)

[New-CMBoundary](#)

[Set-CMBoundary](#)

[Get-CMBoundaryGroup](#)

Get-CMBoundaryGroup

Get-CMBoundaryGroup

Gets a boundary group.

Syntax

Parameter Set: SearchByName

```
Get-CMBoundaryGroup [-Name <String> ] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Get-CMBoundaryGroup -Id <String[]> [ <CommonParameters>]
```

Detailed Description

The **Get-CMBoundaryGroup** cmdlet gets a boundary group. A boundary group is a collection of boundaries.

You can use boundary groups to manage network locations. You must assign boundaries to boundary groups before you can use the boundary group. Boundary groups enable client computers to find a primary site for client assignment, which is referred to as automatic site assignment, and a list of available site systems that have content. For more information about boundaries, see [Planning for Boundaries and Boundary Groups in Configuration Manager](http://go.microsoft.com/fwlink/?LinkId=266225) (<http://go.microsoft.com/fwlink/?LinkId=266225>) on TechNet.

Parameters

-Id<String[]>

Specifies an array of identifiers (IDs) for one or more boundary groups.

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

-Name<String>

Specifies the name for a boundary group.

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

<CommonParameters>

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

Examples

Example 1: Get a boundary group that is specified by its identifier

This command gets a boundary group that is specified by the identifier 1600231.

```
PS C:\> Get-CMBoundaryGroup -Id "1600231"
```

```
CreatedBy:          Contoso\ENarvaez
CreatedOn:          5/17/2012 06:01:29 AM
DefaultSiteCode:
Description:
GroupID:            1600231
MemberCount:       80
ModifiedBy:
ModifiedOn:
Name:               BGroup01
SiteSystemCount:   0
```

Example 2: Get multiple boundary groups that are specified by name

This command gets multiple boundary groups that are specified by the names BGroup01, BGroup02, and BGroup03.

```
PS C:\> Get-CMBoundaryGroup -Name "BGroup01", "BGroup02", "BGroup03"
```

```
CreatedBy:          Contoso\ENarvaez
CreatedOn           5/17/2012 07:13:02 AM
DefaultSiteCode:
Description:
GroupID:            1600231
MemberCount:       80
ModifiedBy:
ModifiedOn:
Name:               BGroup01
SiteSystemCount:   0
CreatedBy:          Contoso\ENarvaez
CreatedOn           7/13/2012 12:24:21 PM
DefaultSiteCode:
Description:
GroupID:            1600246
MemberCount:       11
ModifiedBy:         Contoso\DChew
ModifiedOn:         9/10/2012 04:32:16 PM
Name:               BGroup02
SiteSystemCount:   0
CreatedBy:          Contoso\DChew
CreatedOn           8/06/2012 09:32:05 AM
DefaultSiteCode:
Description:
GroupID:            1600249
MemberCount:       96
ModifiedBy:         Contoso\EDaugherty
ModifiedOn:         9/14/2012 10:11:36 AM
Name:               BGroup03
SiteSystemCount:   0
```

Related topics

[New-CMBoundaryGroup](#)

[Remove-CMBoundaryGroup](#)

[Set-CMBoundaryGroup](#)

Get-CMCategory

Get-CMCategory

Gets configuration categories in Configuration Manager.

Syntax

Parameter Set: GetCategoryByName

```
Get-CMCategory [-CategoryType {UserCategories | BaselineCategories | DriverCategories | AppCategories | GlobalCondition | CatalogCategories} ] [-Name <String[]> ] [ <CommonParameters>]
```

Parameter Set: GetCategoryById

```
Get-CMCategory [-Id <String[]> ] [ <CommonParameters>]
```

Detailed Description

The **Get-CMCategory** cmdlet gets configuration categories in Microsoft System Center 2012 Configuration Manager. Configuration categories offer an optional method of sorting and filtering configuration baselines and configuration items in System Center 2012 Configuration Manager and Configuration Manager reports.

Parameters

-CategoryType<CategoryType>

Specifies a category type. Valid values are:

- BaselineCategories
- DriverCategories
- AppCategories
- GlobalCondition
- CatalogCategories

The acceptable values for this parameter are:

UserCategories	
BaselineCategories	
DriverCategories	

AppCategories	
GlobalCondition	
CatalogCategories	

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

-Id<String[]>

Specifies an array of IDs of configuration categories.

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

-Name<String[]>

Specifies an array of names of configuration categories.

Aliases	LocalizedCategoryInstanceName
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?

false

<CommonParameters>

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

Examples

Example 1: Get configuration categories by using a name

This command gets configuration driver categories in Configuration Manager that have the name NewLaptopDriverSet.

```
PS C:\> Get-CMCategory -CategoryType "DriverCategories" -Name "NewLaptopDriverSet"
```

Related topics

[New-CMCategory](#)

[Remove-CMCategory](#)

Get- CMClientAuthCertificateProfileConfigurationItem

Get-CMClientAuthCertificateProfileConfigurationItem

Gets a certificate profile.

Syntax

Parameter Set: SearchByName

```
Get-CMClientAuthCertificateProfileConfigurationItem [-Name <String[]> ] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Get-CMClientAuthCertificateProfileConfigurationItem -Id <String[]> [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Get-CMClientAuthCertificateProfileConfigurationItem -InputObject <IResultObject> [ <CommonParameters>]
```

Detailed Description

The **Get-CMClientAuthCertificateProfileConfigurationItem** cmdlet gets a certificate profile. Client computers use certificate profiles to authenticate when they use services such as a virtual private network (VPN) or a wireless network.

Parameters

-Id<String[]>

Specifies an array of IDs of certificate profiles.

Aliases	CId
Required?	true
Position?	named
Default Value	none

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

-InputObject<IResultObject>

Specifies a certificate profile object.

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

-Name<String[]>

Specifies an array of names of certificate profiles.

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

<CommonParameters>

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

Examples

Example 1: Get a certificate profile

This command gets the certificate profile that has the specified ID.

PS C:\> Get-CMClientAuthCertificateProfileConfigurationItem -Id "16777568"

Related topics

[Copy-CMClientAuthCertificateProfileConfigurationItem](#)

[New-CMClientAuthCertificateProfileConfigurationItem](#)

[Remove-CMClientAuthCertificateProfileConfigurationItem](#)

[Set-CMClientAuthCertificateProfileConfigurationItem](#)

Get-CMClientOperations

Get-CMClientOperations

Gets all client notification tasks.

Syntax

```
Get-CMClientOperations [ <CommonParameters>]
```

Detailed Description

The **Get-CMClientOperations** cmdlet gets all client notification tasks. Client notification tasks trigger the client agent to perform a specific action on demand without waiting for the client agent to poll for a new policy.

Parameters

<CommonParameters>

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

Examples

Example 1: Get client notification tasks

This command gets all client notification tasks.

```
PS C:\> Get-CMClientOperations
```

Get-CMClientPushInstallation

Get-CMClientPushInstallation

Gets an object that installs a Configuration Manager client by using client push.

Syntax

Parameter Set: SearchByNameMandatory

```
Get-CMClientPushInstallation -SiteSystemServerName <String[]> [ <CommonParameters>]
```

Parameter Set: SearchBySiteCodeMandatory

```
Get-CMClientPushInstallation -SiteCode <String> [-SiteSystemServerName <String[]> ] [ <CommonParameters>]
```

Detailed Description

The **Get-CMClientPushInstallation** cmdlet gets an object that installs a Microsoft System Center 2012 Configuration Manager client by using client push. A client push installation installs client software on computers that System Center 2012 Configuration Manager discovered. When you configure client push installation for a site, the client installation automatically runs on the computers that System Center 2012 Configuration Manager discovered within the site's configured boundaries when those boundaries are part of a boundary group. You can also start a client push installation by running the Client Push Installation Wizard for a specific collection or resource within a collection.

For more information about how to install clients, see [How to Install Clients on Windows Computers in Configuration Manager](http://go.microsoft.com/fwlink/?LinkId=247203) (<http://go.microsoft.com/fwlink/?LinkId=247203>) on TechNet.

Parameters

-SiteCode<String>

Specifies an array of site codes that identify sites on which Configuration Manager installs the client.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

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

-SiteSystemServerName<String[]>

Specifies an array of names of site system servers. Site system servers contain roles that define the operations that each site can perform.

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

<CommonParameters>

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

Examples

Example 1: Get a client push installation

This command gets the client push installation for the site system server named CMClientPushInstallationPoint.Western.Contoso.com.

```
PS C:\> Get-CMClientPushInstallation -SiteSystemServerName  
"CMClientPushInstallationPoint.Western.Contoso.com"
```

Related topics

[Set-CMClientPushInstallation](#)

Get-CMClientSetting

Get-CMClientSetting

Gets client settings.

Syntax

Parameter Set: SearchByName

```
Get-CMClientSetting [-Name <String> ] [-SecuredScopeNames <String> ] [-Setting  
{BackgroundIntelligentTransfer | ClientPolicy | Cloud | ComplianceSettings | ComputerAgent |  
ComputerRestart | EndpointProtection | HardwareInventory | MeteredNetwork | MobileDevice |  
NetworkAccessProtection | PowerManagement | RemoteTools | SoftwareDeployment |  
SoftwareInventory | SoftwareMetering | SoftwareUpdates | StateMessaging |  
UserAndDeviceAffinity} ] [-Type {Default | Device | User} ] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Get-CMClientSetting -Id <String[]> [-SecuredScopeNames <String> ] [-Setting  
{BackgroundIntelligentTransfer | ClientPolicy | Cloud | ComplianceSettings | ComputerAgent |  
ComputerRestart | EndpointProtection | HardwareInventory | MeteredNetwork | MobileDevice |  
NetworkAccessProtection | PowerManagement | RemoteTools | SoftwareDeployment |  
SoftwareInventory | SoftwareMetering | SoftwareUpdates | StateMessaging |  
UserAndDeviceAffinity} ] [-Type {Default | Device | User} ] [ <CommonParameters>]
```

Detailed Description

The **Get-CMClientSetting** cmdlet gets a customized collection of client settings.

For more information about client settings, see [About Client Settings in Configuration Manager](http://go.microsoft.com/fwlink/?LinkId=266226) (<http://go.microsoft.com/fwlink/?LinkId=266226>) on TechNet.

Parameters

-Id<String[]>

Specifies an array of identifiers for one or more collections of client settings.

Aliases	SettingsId
Required?	true
Position?	named

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

-Name<String>

Specifies a name for customized client settings.

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

-SecuredScopeNames<String>

Specifies the name of security scopes. A security scope can be Default or the name of a custom-created security scope.

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

-Setting<SettingType[]>

Specifies an array of setting types for one or more collections of client settings.

The acceptable values for this parameter are:

BackgroundIntelligentTransfer	
-------------------------------	--

ClientPolicy	
Cloud	
ComplianceSettings	
ComputerAgent	
ComputerRestart	
EndpointProtection	
HardwareInventory	
MeteredNetwork	
MobileDevice	
NetworkAccessProtection	
PowerManagement	
RemoteTools	
SoftwareDeployment	
SoftwareInventory	
SoftwareMetering	
SoftwareUpdates	
StateMessaging	
UserAndDeviceAffinity	

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

-Type<Types>

Specifies the type of customized settings. Valid values are 1 (device) or 2 (user).

The acceptable values for this parameter are:

Default	
Device	
User	

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

<CommonParameters>

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

Examples

Example 1: Get a collection of customized client settings that is specified by its name

This command gets client settings that have the specified name.

```
PS C:\> Get-CMClientSetting -Name "Windows 8 Client Computers Settings"
```

```
AgentConfigurations: {}
AssignmentCount:    0
CreatedBy:          Contoso\DChew
DateCreated:        8/04/2012 4:40:03 PM
DateModified:       8/04/2012 4:40:03 PM
Description:         Windows 8 Client Computers Settings
Enabled:             False
FeatureType:        1
Flags:               0
LastModifiedBy:     Contoso\DChew
Name:                Win08ClientSettings
Priority:            0
```

SecuredScopeNames: {Default}
Settings ID: 16777220
Type: 1
UniqueID: {0CCA6700-AE5E-4949-8FBC-AA6719775CC3}

Related topics

[New-CMClientSetting](#)

[Remove-CMClientSetting](#)

[Set-CMClientSetting](#)

Get-CMClientStatusSetting

Get-CMClientStatusSetting

Gets client status settings.

Syntax

```
Get-CMClientStatusSetting [ <CommonParameters> ]
```

Detailed Description

The **Get-CMClientStatusSetting** cmdlet gets client status settings for the local computer. These settings determine the data collection intervals for individual client monitoring activities.

For more information about client settings, see [About Client Settings in Configuration Manager](http://go.microsoft.com/fwlink/?LinkId=266226) (<http://go.microsoft.com/fwlink/?LinkId=266226>) on TechNet.

Parameters

<CommonParameters>

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

Examples

Example 1: Get client status settings for the local computer

This command gets client status settings for the local computer.

```
PS C:\> Get-CMClientStatusSetting
```

```
ADRetrieving Schedule :  
CleanUpInterval       : 7  
DDRInactiveInterval   : 3  
HWInactiveInterval    : 4  
NeedADLastLogonTime   :  
PolicyInactiveInterval : 3
```

SettingsID : 1
StatusInactiveInterval : 6
SWInactiveInterval : 5

Related topics

[Set-CMClientStatusSetting](#)

[Update-CMClientStatus](#)

Get-CMClientStatusUpdateSchedule

Get-CMClientStatusUpdateSchedule

Gets a schedule interval of the client status update task.

Syntax

```
Get-CMClientStatusUpdateSchedule [ <CommonParameters>]
```

Detailed Description

The **Get-CMClientStatusUpdateSchedule** cmdlet gets a schedule interval of the client status update task.

Parameters

<CommonParameters>

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

Examples

Example 1: Gets a client status update schedule

This command gets a client status update schedule.

```
PS C:\> Get-CMClientStatusUpdateSchedule
```

```
Interval Unit
```

```
----- ----
```

```
1 Days
```

Related topics

[Set-CMClientStatusUpdateSchedule](#)

[Get-CMClientStatusSetting](#)

[Set-CMClientStatusSetting](#)



Get-CMCloudDistributionPoint

Get-CMCloudDistributionPoint

Gets cloud-based distribution points.

Syntax

Parameter Set: SearchByName

```
Get-CMCloudDistributionPoint [-Name <String> ] [ <CommonParameters>]
```

Parameter Set: SearchByGroup

```
Get-CMCloudDistributionPoint -DistributionPointGroup <IResultObject> [ <CommonParameters>]
```

Parameter Set: SearchByGroupId

```
Get-CMCloudDistributionPoint -DistributionPointGroupId <String> [ <CommonParameters>]
```

Parameter Set: SearchByGroupName

```
Get-CMCloudDistributionPoint -DistributionPointGroupName <String> [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Get-CMCloudDistributionPoint -Id <String[]> [ <CommonParameters>]
```

Detailed Description

The **Get-CMCloudDistributionPoint** cmdlet gets one or more cloud-based distribution points in Microsoft System Center 2012 Configuration Manager.

In System Center 2012 Configuration Manager, you can use a cloud service in Windows Azure to host a distribution point for storing files to download to clients. You can send packages and apps to and host packages and apps in cloud distribution points.. For more information about cloud distribution points, see [Planning for Content Management in Configuration Manager](http://go.microsoft.com/fwlink/?LinkId=266223) (<http://go.microsoft.com/fwlink/?LinkId=266223>) on TechNet.

You can use the **Get-CMCloudDistributionPoint** cmdlet to get distribution points to use with other cmdlets. For instance, you might want to get a distribution point and then use the **Stop-CMCloudDistributionPoint** cmdlet to suspend it.

Parameters

-DistributionPointGroup<IResultObject>

Specifies a **CMDistributionPointGroup** object. To obtain a **CMDistributionPointGroup** object, use the **Get-CMDistributionPointGroup** cmdlet.

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

-DistributionPointGroupId<String>

Specifies the ID of a distribution point group.

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

-DistributionPointGroupName<String>

Specifies the name of a distribution point group.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

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

-Id<String[]>

Specifies an array of identifiers for cloud distribution points. You can use a comma separated list.

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

-Name<String>

Specifies the name of a cloud distribution point.

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

<CommonParameters>

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

Examples

Example 1: Get all cloud distribution points

This command gets all the cloud distribution points for a System Center 2012 Configuration Manager.

```
PS C:\> Get-CMCloudDistributionPoint
```

Example 2: Get a cloud distribution point using a name

This command gets a distribution point named West01.

```
PS C:\> Get-CMCloudDistributionPoint -Name "West01"
```

Example 3: Get a cloud distribution point using an ID

This command gets a distribution point with the specified identifier.

```
PS C:\> Get-CMCloudDistributionPoint -Id "16777230"
```

Related topics

[New-CMCloudDistributionPoint](#)

[Remove-CMCloudDistributionPoint](#)

[Set-CMCloudDistributionPoint](#)

[Start-CMCloudDistributionPoint](#)

[Stop-CMCloudDistributionPoint](#)

Get- CMCollectionMembershipEvaluationComponent

Get-CMCollectionMembershipEvaluationComponent

Gets how often Configuration Manager evaluates collection membership.

Syntax

Parameter Set: SearchByNameMandatory

```
Get-CMCollectionMembershipEvaluationComponent -SiteSystemServerName <String[]> [  
<CommonParameters>]
```

Parameter Set: SearchBySiteCodeMandatory

```
Get-CMCollectionMembershipEvaluationComponent -SiteCode <String> [-SiteSystemServerName  
<String[]> ] [ <CommonParameters>]
```

Detailed Description

The **Get-CMCollectionMembershipEvaluationComponent** cmdlet gets the value for how often Microsoft System Center 2012 Configuration Manager evaluates collections. System Center 2012 Configuration Manager queries the database at a regular interval to check for changes in collection membership. You can specify which value to get by site server name or site code.

Parameters

-SiteCode<String>

Specifies a site codes for a Configuration Manager site.

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

-SiteSystemServerName<String[]>

Specifies an array of names for Configuration Manager servers.

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

<CommonParameters>

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

Examples

Example 1: Get an evaluation period for a site code

This command gets the evaluation frequency for collection membership for the specified site code.

```
PS C:\> Get-CMCollectionMembershipEvaluationComponent -SiteCode "CM4"
```

Example 2: Get an evaluation period for a system

This command gets the evaluation frequency for the server named CM01.West01.Contoso.com.

```
PS C:\> Get-CMCollectionMembershipEvaluationComponent -SiteSystemServerName  
"CM01.West01.Contoso.com"
```

Related topics

[Set-CMCollectionMembershipEvaluationComponent](#)

Get-CMComponentStatusMessage

Get-CMComponentStatusMessage

Gets component status messages in Configuration Manager.

Syntax

Parameter Set: SearchByName

```
Get-CMComponentStatusMessage -ViewingPeriod <DateTime> [-ComponentName <String> ] [-ComputerName <String> ] [-Severity {All | Error | Information | Warning} ] [-SiteCode <String> ] [ <CommonParameters>]
```

Detailed Description

The **Get-CMComponentStatusMessage** cmdlet gets component status messages for a specified period.

Microsoft System Center 2012 Configuration Manager indicates whether operations succeed or fail and include other information in component status messages. Threads or processes send component status messages to System Center 2012 Configuration Manager sites, identified by site codes.

You can define which messages to get by the severity of the message, the component that created the message, the computer that hosts that component, or the System Center 2012 Configuration Manager server that receives the message. You must specify a viewing period, as a TimeSpan object.

Parameters

-ComponentName<String>

Specifies the name of a thread or process. A thread or process sends a component status message.

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

-ComputerName<String>

Specifies the name of a computer. A computer hosts a component that sends a status message.

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

-Severity<Severity>

Specifies the severity of status messages. Valid values are:

- ALL
- Error
- Information
- Warning

The acceptable values for this parameter are:

All	
Error	
Information	
Warning	

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

-SiteCode<String>

Specifies an array of a site codes for Configuration Manager sites.

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

-ViewingPeriod<DateTime>

Specifies a **DateTime** object. To obtain a **DateTime** object, use the **Get-Date** cmdlet. For more information, type `Get-Help Get-Date`. The cmdlet limits component status messages to this time period.

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

<CommonParameters>

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

Examples

Example 1: Get critical messages for a site

This command gets component status messages for the specified viewing period for the System Center 2012 Configuration Manager site that has the site code CM1. The command gets only critical messages.

```
PS C:\> Get-CMComponentStatusMessage -ViewingPeriod "2/1/2013 12:00 AM" -Severity Critical -SiteCode "CM1"
```

Related topics

[Get-CMComponentStatusSetting](#)

Get-CMComponentStatusSetting

Get-CMComponentStatusSetting

Gets Configuration Manager component status settings.

Syntax

Parameter Set: SearchByName

```
Get-CMComponentStatusSetting [-ComponentName <String> ] [-SiteCode <String> ] [-SiteSystemName <String[]> ] [ <CommonParameters>]
```

Detailed Description

The **Get-CMComponentStatusSetting** cmdlet gets component status setting objects. These objects contain thresholds for Microsoft System Center 2012 Configuration Manager component status messages.

Parameters

-ComponentName<String>

Specifies a name of a thread or process. A thread or process sends a component status message. You can use wildcards.

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

-SiteCode<String>

Specifies an array of a site codes for Configuration Manager sites.

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

-SiteSystemName<String[]>

Specifies an array of names of computers. A computer hosts a component that sends a status message. You can use wildcards.

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

<CommonParameters>

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

Examples

Example 1: Get settings for components

This command gets status setting objects for the site that has the site code CAS for components with names that begin with SMS_REPL.

```
PS C:\> Get-CMComponentStatusSetting -SiteCode "CAS" -ComponentName SMS_REPL*
```

Example 2: Get settings for components on specified systems

This command gets status setting objects for systems with names that begin with VM for components with names that begin with SMS_REPL.

```
PS C:\> Get-CMComponentStatusSetting -SiteSystemName VM* -ComponentName SMS_REPL*
```

Related topics

[Get-CMComponentStatusMessage](#)

Get-CMComputerAssociation

Get-CMComputerAssociation

Gets Configuration Manager computer associations.

Syntax

Parameter Set: SearchByName

```
Get-CMComputerAssociation [-DestinationComputer <String> ] [-SourceComputer <String> ] [ <CommonParameters> ]
```

Parameter Set: SearchByIdMandatory

```
Get-CMComputerAssociation -MigrationId <String> [ <CommonParameters> ]
```

Detailed Description

The **Get-CMComputerAssociation** cmdlet gets computer associations. Microsoft System Center 2012 Configuration Manager uses a computer association to migrate user state and settings as part of operating system deployment. You can specify a source computer, a destination computer, or both. You can also use an ID to specify a computer association.

Parameters

-DestinationComputer<String>

Specifies the name of a destination computer.

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

-MigrationId<String>

Specifies the ID of a computer association.

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

-SourceComputer<String>

Specifies the name of a source computer.

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

<CommonParameters>

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

Examples

Example 1: Get all computer associations

This command gets all the computer associations for System Center 2012 Configuration Manager.

```
PS C:\> Get-CMComputerAssociation
```

Example 2: Get computer associations for a destination computer

This command gets all the computer associations for the destination computer named West155.

```
PS C:\> Get-CMComputerAssociation -DestinationComputer "West155"
```

Example 3: Get a computer association by using an ID

This command gets the computer association that has the ID MID1207.

```
PS C:\> Get-CMComputerAssociation -MigrationId "MID1207"
```

Related topics

[New-CMComputerAssociation](#)

[Remove-CMComputerAssociation](#)

[Set-CMComputerAssociation](#)

Get-CMConfigurationItem

Get-CMConfigurationItem

Gets Configuration Manager configuration items.

Syntax

Parameter Set: SearchByName

```
Get-CMConfigurationItem [-Name <String[]> ] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Get-CMConfigurationItem -Id <String[]> [ <CommonParameters>]
```

Detailed Description

The **Get-CMConfigurationItem** cmdlet gets configuration item objects in Microsoft System Center 2012 Configuration Manager. You can use this cmdlet to get items for other cmdlets to use. For instance, you might get configuration items so you can use the **Set-CMConfigurationItem** to change settings on them.

Configuration items contain one or more settings, along with compliance rules. Items usually define a unit of configuration you want to. For more information about configuration items, see [Introduction to Compliance Settings in Configuration Manager](http://go.microsoft.com/fwlink/?LinkId=211014) (<http://go.microsoft.com/fwlink/?LinkId=211014>) on TechNet.

Parameters

-Id<String[]>

Specifies an array of identifiers for one or more configuration items. You can use a comma separated list.

Aliases	CIId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

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

-Name<String[]>

Specifies an array of names of configuration items. You can use a comma separated list.

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

<CommonParameters>

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

Examples

Example 1: Get an item using a name

This command gets a configuration item named ConfigItem76.

```
PS C:\> Get-CMConfigurationItem -Name "ConfigItem76"
```

Example 2: Get an item to use with another cmdlet

The first command gets a configuration item with the specified identifier and stores it in the \$CIObj variable.

The second command removes the item in the \$CIObj variable.

```
PS C:\> $CIObj=Get-CMConfigurationItem -Id "16777568"  
PS C:\> Remove-CMConfigurationItem -InputObject $CIObj
```

Related topics

[Export-CMConfigurationItem](#)

[Get-CMConfigurationItemXMLDefinition](#)

[Import-CMConfigurationItem](#)

[New-CMConfigurationItem](#)

[Remove-CMConfigurationItem](#)

[Set-CMConfigurationItem](#)

[Get-CMConfigurationItemHistory](#)

Get-CMConfigurationItemHistory

Get-CMConfigurationItemHistory

Gets the previous versions of a configuration item in Configuration Manager.

Syntax

Parameter Set: SearchByIdMandatory

```
Get-CMConfigurationItemHistory -Id <String[]> [-Revision <Int32> ] [ <CommonParameters>]
```

Parameter Set: SearchByNameMandatory

```
Get-CMConfigurationItemHistory -Name <String[]> [-Revision <Int32> ] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Get-CMConfigurationItemHistory -InputObject <IResultObject> [-Revision <Int32> ] [ <CommonParameters>]
```

Detailed Description

The **Get-CMConfigurationItemHistory** cmdlet gets the previous versions of a configuration item.

Microsoft System Center 2012 Configuration Manager updates configuration items based on configuration management, software updates management, and operating system deployment. System Center 2012 Configuration Manager stores the previous version of the item. The server removes previous versions, by default, when the data is more than 90 days old.

This cmdlet gets the history of an item for a specified item name. This cmdlet also gets the history for a specified revision of an item.

Parameters

-Id<String[]>

Specifies an identifier for a configuration item revision.

Aliases	CIId
Required?	true
Position?	named
Default Value	none

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

-InputObject<IResultObject>

Specifies a configuration item object. To obtain a configuration item object, use the **Get-
CMConfigurationItem** cmdlet.

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

-Name<String[]>

Specifies an array of names of configuration items.

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

-Revision<Int32>

Specifies the version of a configuration item as an integer.

Aliases	none
Required?	false
Position?	named

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

<CommonParameters>

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

Examples

Example 1: Get item history by using a name

This command gets the history for a configuration item named CMCI07.

```
PS C:\> Get-CMConfigurationItemHistory -Name "CMCI07"
```

Example 2: Get item history by using an ID

This command gets the previous version of a configuration item with the specified ID.

```
PS C:\> Get-CMConfigurationItemHistory -Id "16777568"
```

Related topics

[Export-CMConfigurationItem](#)

[Get-CMConfigurationItem](#)

[Get-CMConfigurationItemXMLDefinition](#)

[Remove-CMConfigurationItem](#)

[Set-CMConfigurationItem](#)

Get-CMConfigurationItemXMLDefinition

Get-CMConfigurationItemXMLDefinition

Gets an XML definition of a configuration item in Configuration Manager.

Syntax

Parameter Set: SearchByIdMandatory

```
Get-CMConfigurationItemXMLDefinition -Id <String[]> [ <CommonParameters>]
```

Parameter Set: SearchByNameMandatory

```
Get-CMConfigurationItemXMLDefinition -Name <String[]> [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Get-CMConfigurationItemXMLDefinition -InputObject <IResultObject> [ <CommonParameters>]
```

Detailed Description

The **Get-CMConfigurationItemXMLDefinition** cmdlet gets an XML definition of a configuration item object as a string. You can specify a configuration item with the configuration item ID, the configuration item name, or using the **Get-CMConfigurationItem** cmdlet.

Parameters

-Id<String[]>

Specifies an array of identifiers of configuration items. You can use a comma separated list.

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

-InputObject<IResultObject>

Specifies a configuration item object. To obtain a configuration item object, use the **Get-
CMConfigurationItem** cmdlet.

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

-Name<String[]>

Specifies an array of names of configuration items. You can use a comma separated list.

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

<CommonParameters>

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

Examples

Example 1: Get XML formatted item using an ID

This command gets a configuration item formatted in XML for the item that has the specified identifier.

```
PS C:\> Get-ConfigurationItemXMLDefinition -Id "16777568"
```

Example 2: Get XML formatted item using a name

This command gets a configuration item formatted in XML for the item named ConfigItem76.

```
PS C:\> Get-CMConfigurationItemXMLDefinition -Name "ConfigItem76"
```

Example 3: Get XML formatted item using a variable

The first command uses the **Get-ConfigurationItem** cmdlet to get a configuration item with the specified ID, and then stores it in the \$CIObj variable.

The second command gets a configuration item formatted in XML for the item stored in \$CIObj.

```
PS C:\> $CIObj=Get-CMConfigurationItem -Id "16777568"
```

```
PS C:\> Get-CMConfigurationItemXMLDefinition -InputObject $CIObj
```

Related topics

[Export-CMConfigurationItem](#)

[Get-CMConfigurationItem](#)

[Remove-CMConfigurationItem](#)

[Set-CMConfigurationItem](#)

[Get-CMConfigurationItemHistory](#)

Get-CMConflictingRecord

Get-CMConflictingRecord

Gets conflicting Configuration Manager record objects.

Syntax

Parameter Set: SearchByName

```
Get-CMConflictingRecord [-Name <String> ] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Get-CMConflictingRecord -Id <String> [ <CommonParameters>]
```

Detailed Description

The **Get-CMConflictingRecord** cmdlet gets one or more conflicting Microsoft System Center 2012 Configuration Manager record objects.

When System Center 2012 Configuration Manager recognizes a new client, it uses hardware information to check whether it previously created a record for that computer. For example, you might have reinstalled the operating system. The previous client record still exists with the same hardware information. If you manually resolve conflicts, you have the option to merge the new record with the existing record, create a new record, or create a record as a blocked record. You can also configure System Center 2012 Configuration Manager to resolve conflicts automatically.

You can use this cmdlet with the [Block-CMConflictingRecord](#) cmdlet or the [Merge-CMConflictingRecord](#) cmdlet. You can get all the outstanding conflicts for System Center 2012 Configuration Manager or specify a conflict by name or by ID.

Parameters

-Id<String>

Specifies an ID for the conflicting records.

Aliases	Smsid
Required?	true
Position?	named
Default Value	none

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

-Name<String>

Specifies a name for the conflicting records.

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

<CommonParameters>

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

Examples

Example 1: Get all conflicting records

This command gets all the unresolved conflicts for System Center 2012 Configuration Manager.

```
PS C:\> Get-CMConflictingRecord
```

Example 2: Get a named conflicting record

This command gets a conflict named CR07.

```
PS C:\> Get-CMConflictingRecord -Name "CR07"
```

Related topics

[Block-CMConflictingRecord](#)

[Merge-CMConflictingRecord](#)



Get-CMDatabaseProperty

Get-CMDatabaseProperty

Gets an object that represents a Configuration Manager database.

Syntax

Parameter Set: SearchBySiteCode

```
Get-CMDatabaseProperty -SiteCode <String> [ <CommonParameters>]
```

Detailed Description

The **Get-CMDatabaseProperty** cmdlet gets an object that represents a Microsoft System Center 2012 Configuration Manager database. Use the site code for a site to specify a database.

When this cmdlet returns a database object in the console, it displays current settings for data compression, Broker port for the computer that runs Microsoft SQL Server, and the length of time that the database keeps data. You can use the **Set-CMDatabaseProperty** cmdlet to change these values.

Parameters

-SiteCode<String>

Specifies the site code for a Configuration Manager site.

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

<CommonParameters>

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

Examples

Example 1: Get a database property

This command gets the database property for the site that has the site code CM2.

```
PS C:\> Get-CMDatabaseProperty -SiteCode "CM2"
```

Key	Value
---	-----
SQL Server Service Broker Port	80
Retention Period	10
IsCompression	0

Related topics

[Set-CMDatabaseProperty](#)

Get-CMDatabaseReplicationLinkProperty

Get-CMDatabaseReplicationLinkProperty

Gets a replication link between a Configuration Manager parent site and child site.

Syntax

Parameter Set: SearchBySiteMandatory

```
Get-CMDatabaseReplicationLinkProperty -ChildSiteCode <String> -ParentSiteCode <String> [  
<CommonParameters>]
```

Detailed Description

The **Get-CMDatabaseReplicationLinkProperty** cmdlet gets a specified replication link between a Microsoft System Center 2012 Configuration Manager parent site and child site.

Database replication for System Center 2012 Configuration Manager sites transfers data and merges changes made in a site database with information stored at other sites in the System Center 2012 Configuration Manager hierarchy. This enables all sites to share the same information.

Parameters

-ChildSiteCode<String>

Specifies a site code for a Configuration Manager site. This parameter refers to the child site.

Aliases	Site2
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ParentSiteCode<String>

Specifies a site code for a Configuration Manager site. This parameter refers to the parent site.

Aliases	Site1
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

<CommonParameters>

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

Examples

Example 1: Get a replication link

This command gets a replication link between specified parent and child sites. You must specify both sites.

```
PS C:\> Get-CMDatabaseReplicationLinkProperty -ChildSiteCode "CM8" -ParentSiteCode "CM1"
```

Related topics

[Set-CMDatabaseReplicationLinkProperty](#)

Get-CMDataBaseReplicationStatus

Get-CMDataBaseReplicationStatus

Gets the status for database replication.

Syntax

Parameter Set: SearchBySite

```
Get-CMDataBaseReplicationStatus [-ChildSiteCode <String> ] [-ParentSiteCode <String> ] [ <CommonParameters> ]
```

Detailed Description

The **Get-CMDatabaseReplicationStatus** cmdlet gets the status of the database replication link for a Microsoft System Center 2012 Configuration Manager parent/child site pair. The cmdlet identifies the sites by site code.

You can specify just the site code or just the name for a parent or child and get all the database replication links for the specified site.

Parameters

-ChildSiteCode<String>

Specifies a site code for a child site.

Aliases	Site2
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ParentSiteCode<String>

Specifies a site code for a parent site.

Aliases	Site1
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

<CommonParameters>

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

Examples

Example 1: Get status using site codes

This command gets the status of a database replication link for the child with a site code CCC and the parent with a site code CCA.

```
PS C:\> Get-CMDataBaseReplicationStatus -ChildSiteCode "CCC" -ParentSiteCode "CCA"
```

Get-CMDeployment

Get-CMDeployment

Gets a summary of Configuration Manager deployments.

Syntax

Parameter Set: SearchByName

```
Get-CMDeployment [-CollectionName <String> ] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Get-CMDeployment -DeploymentId <String> [ <CommonParameters>]
```

Detailed Description

The **Get-CMDeployment** cmdlet gets summary information for one or more deployments in Microsoft System Center 2012 Configuration Manager. The cmdlet gets summary information about application, Software Update Management (SUM), or classic program deployments in System Center 2012 Configuration Manager.

Parameters

-CollectionName<String>

Specifies an array of names of Configuration Manager collections to which the deployment is applied.

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

-DeploymentId<String>

Specifies the ID of a deployment.

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

<CommonParameters>

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

Examples

Example 1: Get deployments by using an ID

This command gets the deployment that has the ID Cm120006.

```
PS C:\> Get-CMDeployment -DeploymentId "Cm120006"
```

Example 2: Get deployments by using a name

This command gets the deployment named CMCollect01.

```
PS C:\> Get-CMDeployment -CollectionName "CMCollect01"
```

Related topics

[Get-CMDeploymentStatus](#)

[Remove-CMDeployment](#)

[Get-CMDeploymentType](#)

[Set-CMDeploymentType](#)

Get-CMDeploymentPackage

Get-CMDeploymentPackage

Gets information about deployment packages on a distribution point.

Syntax

Parameter Set: SearchByName

```
Get-CMDeploymentPackage -DistributionPointName <String> [-DeploymentPackageName <String[]> ]  
[ <CommonParameters>]
```

Detailed Description

The **Get-CMDeploymentPackage** cmdlet gets information about one or more deployment packages on a distribution point. If you do not specify the *DeploymentPackageName* parameter, Microsoft System Center 2012 Configuration Manager returns all the deployment packages on the distribution point that you specify.

A deployment package is a System Center 2012 Configuration Manager object that contains the content files and instructions for distributing programs, software updates, boot images, operating system images, and drivers to System Center 2012 Configuration Manager clients.

Parameters

-DeploymentPackageName<String[]>

Specifies an array of names of distribution points that are associated with deployment packages. If you do not specify this parameter, the cmdlet returns status information about all deployment packages on the distribution point.

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

-DistributionPointName<String>

Specifies an array of names of deployment packages.

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

<CommonParameters>

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

Examples

Example 1: Get all deployment packages for a distribution point

This command gets all deployment packages that are distributed to clients from the distribution point named CMCEN-DIST02.TSQA.CORP.CONTOSCO.COM.

```
PS C:\> Get-CMDeploymentPackage -DistributionPointName "CMCEN-DIST02.TSQA.CORP.CONTOSCO.COM"
```

Example 2: Gets a deployment package for a distribution point

This command gets the deployment package named Depack01 that is distributed to clients from the distribution point named CMCEN-DIST02.TSQA.CORP.CONTOSCO.COM.

```
PS C:\> Get-CMDeploymentPackage -DistributionPointName "CMCEN-DIST02.TSQA.CORP.CONTOSCO.COM" -DeploymentPackageName "Depack01"
```

Related topics

[Get-CMDeployment](#)

[Get-CMDeploymentStatus](#)

[Get-CMDeploymentType](#)

[Invoke-CMDeploymentSummarization](#)

Get-CMDeploymentStatus

Get-CMDeploymentStatus

Gets the status of classic software distribution deployments.

Syntax

Parameter Set: SearchByName

```
Get-CMDeploymentStatus [-Name <String[]> ] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Get-CMDeploymentStatus -Id <String[]> [ <CommonParameters>]
```

Detailed Description

The **Get-CMDeploymentStatus** cmdlet gets the status of one or more classic software distribution deployments. A classic software distribution is a legacy software distribution program on a client.

Parameters

-Id<String[]>

Specifies an array of IDs of deployments.

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

-Name<String[]>

Specifies an array of names of deployment packages.

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

<CommonParameters>

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

Examples

Example 1: Get the status of a deployment

This command gets the status of a deployment that is distributed to System Center 2012 Configuration Manager clients by using the deployment package named Depack01.

```
PS C:\> Get-CMDeploymentStatus -Name "Depack01"
```

Related topics

[Get-CMDeploymentType](#)

[Get-CMDeploymentPackage](#)

[Invoke-CMDeploymentSummarization](#)

[Get-CMDeploymentType](#)

[Remove-CMDeployment](#)

Get-CMDeploymentType

Get-CMDeploymentType

Gets the deployment type of an application.

Syntax

Parameter Set: SearchByName

```
Get-CMDeploymentType -ApplicationName <String> [-DeploymentTypeName <String> ] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Get-CMDeploymentType -ApplicationName <String> -DeploymentTypeId <String> [ <CommonParameters>]
```

Detailed Description

The **Get-CMDeploymentType** cmdlet gets the deployment type of an application. A deployment type is contained within an application and contains the information that Microsoft System Center 2012 Configuration Manager requires to install software. A deployment type also contains rules that specify if and how the software is deployed.

Parameters

-ApplicationName<String>

Specifies the name of an application that is associated with the deployment type.

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

-DeploymentTypeId<String>

Specifies the ID of a deployment type.

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

-DeploymentTypeName<String>

Specifies the name of a deployment type.

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

<CommonParameters>

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

Examples

Example 1: Get the deployment type of an application

This command gets the deployment type for the application named CenterApp.

```
PS C:\> Get-CMDeploymentType -ApplicationName "CenterApp"
```

Example 2: Get the deployment type of an application by using a name

This command gets the deployment type for the application named CenterApp that has a deployment type named InterDept - Windows app package (.appx file).

```
PS C:\> Get-CMDeploymentType -ApplicationName "CenterApp" -DeploymentTypeName "InterDept - Windows app package (.appx file)"
```

Example 3: Get the deployment type of an application by using an ID

This command gets the deployment type for the application named CenterApp that has a deployment type that has the ID 16777457.

```
PS C:\> Get-CMDeploymentType -ApplicationName "CenterApp" -DeploymentTypeID "16777457"
```

Related topics

[Get-CMDeployment](#)

[Add-CMDeploymentType](#)

[Remove-CMDeploymentType](#)

[Set-CMDeploymentType](#)

[Get-CMDeploymentStatus](#)

[Get-CMDeploymentPackage](#)

Get-CMDevice

Get-CMDevice

Gets a device of the Configuration Manager hierarchy.

Syntax

Parameter Set: ByName

```
Get-CMDevice [-Name <String> ] [ <CommonParameters>]
```

Parameter Set: ById

```
Get-CMDevice -Id <String> [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Get-CMDevice -CollectionId <String> [-Name <String> ] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatoryForViewInfectedClients

```
Get-CMDevice -ThreatId <String> [-CollectionId <String> ] [ <CommonParameters>]
```

Parameter Set: SearchByNameMandatory

```
Get-CMDevice -CollectionName <String> [-Name <String> ] [ <CommonParameters>]
```

Parameter Set: SearchByNameMandatoryForViewInfectedClients

```
Get-CMDevice -ThreatName <String> [-CollectionId <String> ] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Get-CMDevice -Collection <IResultObject> [-Name <String> ] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatoryForViewInfectedClients

```
Get-CMDevice -Threat <IResultObject> [-CollectionId <String> ] [ <CommonParameters>]
```

Detailed Description

The **Get-CMDevice** cmdlet retrieves a device object that can be added to a collection.

To add a device to a collection use the [Add-CMDeviceCollectionDirectMembershipRule](#) cmdlet. For more information about Configuration Manager collections, see [Introduction to Collections in Configuration Manager](#) (<http://go.microsoft.com/fwlink/p/?LinkID=259433>) on TechNet.

Parameters

-Collection<IResultObject>

Specifies an object that represents the device collection. To obtain a device collection object, use the **Get-CMDeviceCollection** cmdlet.

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

-CollectionId<String>

Specifies an ID for a collection in Configuration Manager.

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

-CollectionName<String>

Specifies a name of a collection in Configuration Manager.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

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

-Id<String>

Specifies the ID of the device.

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

-Name<String>

Specifies the name of the device.

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

-Threat<IResultObject>

Specifies an object that represents a threat.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

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

-ThreatId<String>

Specifies an ID for a threat.

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

-ThreatName<String>

Specifies a name for a threat.

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

<CommonParameters>

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

Examples

Example 1: Get a device

This command gets the device object named CMCEN-DIST02.

```
PS C:\> Get-CMDevice -Name "CMCEN-DIST02"
```

Related topics

[Remove-CMDevice](#)

[Approve-CMDevice](#)

[Unblock-CMDevice](#)

[Block-CMDevice](#)

[Get-CMDeviceCollection](#)

Get-CMDeviceCollection

Get-CMDeviceCollection

Gets one or more device collections in the Configuration Manager hierarchy.

Syntax

Parameter Set: SearchByName

```
Get-CMDeviceCollection [-Name <String> ] [ <CommonParameters>]
```

Parameter Set: SearchByDPGroupId

```
Get-CMDeviceCollection -DistributionPointGroupId <String> [ <CommonParameters>]
```

Parameter Set: SearchByDPGroupName

```
Get-CMDeviceCollection -DistributionPointGroupName <String> [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Get-CMDeviceCollection -CollectionId <String> [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Get-CMDeviceCollection -DistributionPointGroup <IResultObject> [ <CommonParameters>]
```

Detailed Description

The **Get-CMDeviceCollection** cmdlet retrieves collections that contain computers or mobile devices. For more information about collections, see [Introduction to Collections in Configuration Manager](http://go.microsoft.com/fwlink/p/?LinkID=259433) (<http://go.microsoft.com/fwlink/p/?LinkID=259433>) on TechNet.

Parameters

-CollectionId<String>

Specifies the ID of the device collection.

Aliases	none
Required?	true
Position?	named
Default Value	none

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

-DistributionPointGroup<IResultObject>

Specifies an object that represents a distribution point group that is associated with the device collections.

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

-DistributionPointGroupId<String>

Specifies the ID of a distribution point group that is associated with the device collections.

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

-DistributionPointGroupName<String>

Specifies the name of a distribution point group that is associated with the device collections.

Aliases	none
Required?	true
Position?	named

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

-Name<String>

Specifies the names of the device collections. If a specific collection is not specified, all device collections in the hierarchy are returned.

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

<CommonParameters>

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

Examples

Example 1: Get a device collection by using an ID

This command gets the device collection that has the ID 9990000D.

```
PS C:\> Get-CMDeviceCollection -CollectionId "9990000D"
```

Related topics

[Export-CMDeviceCollection](#)

[Import-CMDeviceCollection](#)

[New-CMDeviceCollection](#)

[Remove-CMDeviceCollection](#)

[Set-CMDeviceCollection](#)

Get-CMDeviceCollectionDirectMembershipRule

Get-CMDeviceCollectionDirectMembershipRule

Gets the direct membership rules of device collections in the Configuration Manager hierarchy.

Syntax

Parameter Set: ByCollectionNameAndResourceName

```
Get-CMDeviceCollectionDirectMembershipRule -CollectionName <String> -ResourceName <String> [  
<CommonParameters>]
```

Parameter Set: ByCollectionIdAndResourceId

```
Get-CMDeviceCollectionDirectMembershipRule -CollectionId <String> -ResourceId <Int32> [  
<CommonParameters>]
```

Parameter Set: ByCollectionIdAndResourceName

```
Get-CMDeviceCollectionDirectMembershipRule -CollectionId <String> -ResourceName <String> [  
<CommonParameters>]
```

Parameter Set: ByCollectionNameAndResourceId

```
Get-CMDeviceCollectionDirectMembershipRule -CollectionName <String> -ResourceId <Int32> [  
<CommonParameters>]
```

Parameter Set: ByCollectionValueAndResourceId

```
Get-CMDeviceCollectionDirectMembershipRule -Collection <IResultObject> -ResourceId <Int32> [  
<CommonParameters>]
```

Parameter Set: ByCollectionValueAndResourceName

```
Get-CMDeviceCollectionDirectMembershipRule -Collection <IResultObject> -ResourceName  
<String> [ <CommonParameters>]
```

Detailed Description

The **Get-CMDeviceCollectionDirectMembershipRule** cmdlet gets the direct membership rules of one or more collections. You can specify the device collections by using their names, IDs, or by specifying an object that represents the collections.

A direct rule lets you explicitly choose the members of the device collection. For more information about collection rules in Microsoft System Center 2012 Configuration Manager, see [Introduction to Collections in Configuration Manager](http://go.microsoft.com/fwlink/p/?LinkID=259433) (<http://go.microsoft.com/fwlink/p/?LinkID=259433>) on TechNet.

Parameters

-Collection<IResultObject>

Specifies a Configuration Manager device collection object. To obtain a device collection object, use the **Get-CMDeviceCollection** cmdlet.

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

-CollectionId<String>

Specifies the ID of the device collection.

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

-CollectionName<String>

Specifies the name of the device collection where the rule is applied.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

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

-ResourceId<Int32>

Specifies the ID of the rule that you want to retrieve. To retrieve all the direct rules for the collections leave this parameter blank.

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

-ResourceName<String>

Specifies the name of the rule that you want to retrieve.

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

<CommonParameters>

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

Examples

Example 1: Get all direct membership rules

This command gets all the direct membership rules of the device collection that has the ID CM0001A.

```
PS C:\> Get-CMDeviceCollectionDirectMembershipRule -CollectionID "CM0001A" -ResourceId  
"Res_94412512"
```

Related topics

[Add-CMDeviceCollectionDirectMembershipRule](#)

[Remove-CMDeviceCollectionDirectMembershipRule](#)

[Get-CMUserCollection](#)

Get- CMDeviceCollectionExcludeMembershipRule

Get-**CMDeviceCollectionExcludeMembershipRule**

Gets the exclude membership rules from one or more device collections in the Configuration Manager hierarchy.

Syntax

Parameter Set: ByCollectionNameAndExcludeCollectionName

```
Get-CMDeviceCollectionExcludeMembershipRule -CollectionName <String> -ExcludeCollectionName <String> [ <CommonParameters>]
```

Parameter Set: ByCollectionIdAndExcludeCollectionId

```
Get-CMDeviceCollectionExcludeMembershipRule -CollectionId <String> -ExcludeCollectionId <String> [ <CommonParameters>]
```

Parameter Set: ByCollectionIdAndExcludeCollectionName

```
Get-CMDeviceCollectionExcludeMembershipRule -CollectionId <String> -ExcludeCollectionName <String> [ <CommonParameters>]
```

Parameter Set: ByCollectionNameAndExcludeCollectionId

```
Get-CMDeviceCollectionExcludeMembershipRule -CollectionName <String> -ExcludeCollectionId <String> [ <CommonParameters>]
```

Parameter Set: ByCollectionValueAndExcludeCollectionId

```
Get-CMDeviceCollectionExcludeMembershipRule -Collection <IResultObject> -ExcludeCollectionId <String> [ <CommonParameters>]
```

Parameter Set: ByCollectionValueAndExcludeCollectionName

```
Get-CMDeviceCollectionExcludeMembershipRule -Collection <IResultObject> -ExcludeCollectionName <String> [ <CommonParameters>]
```

Detailed Description

The **Get-**CMDeviceCollectionExcludeMembershipRule**** cmdlet retrieves the rules that exclude the members of another collection from the device collections where the rule is applied. You can specify the device collections where the rule is applied by using their names, IDs, or by specifying an object that represents the collections. You can specify the collection whose members are excluded by using its name, ID, or an object that represents the collection.

Microsoft System Center 2012 Configuration Manager dynamically updates the membership of the device collection on a schedule if the membership of the excluded collection changes. For more information about membership rules, see [Introduction to Collections in Configuration Manager](http://go.microsoft.com/fwlink/p/?LinkID=259433) (<http://go.microsoft.com/fwlink/p/?LinkID=259433>) on TechNet.

Parameters

-Collection<IResultObject>

Specifies a Configuration Manager device collection object. To obtain a device collection object, use the **Get-CMDeviceCollection** cmdlet.

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

-CollectionId<String>

Specifies the IDs of the device collections where the rule is applied.

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

-CollectionName<String>

Specifies the name of the device collection where the rule is applied.

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

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

-ExcludeCollectionId<String>

Specifies the ID of the collection whose members are excluded in the device collection.

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

-ExcludeCollectionName<String>

Specifies the name of the collection whose members are excluded from the device collections.

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

<CommonParameters>

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

Examples

Example 1: Get the exclude membership rules from a device collection

This command gets the rules that exclude the members of the collection that has the ID SMSDM001 from the device collection that has the ID 9990000D.

```
PS C:\> Get-CMDeviceCollectionExcludeMembershipRule -CollectionId "9990000D" -  
ExcludeCollectionId "SMSDM001"
```

Related topics

[Add-CMDeviceCollectionExcludeMembershipRule](#)

[Remove-CMDeviceCollectionExcludeMembershipRule](#)

Get- CMDeviceCollectionIncludeMembershipRule

Get-**CMDeviceCollectionIncludeMembershipRule**

Gets the include membership rules from one or more device collections in the Configuration Manager hierarchy.

Syntax

Parameter Set: ByCollectionNameAndIncludeCollectionName

```
Get-CMDeviceCollectionIncludeMembershipRule -CollectionName <String> -IncludeCollectionName <String> [ <CommonParameters>]
```

Parameter Set: ByCollectionIdAndIncludeCollectionId

```
Get-CMDeviceCollectionIncludeMembershipRule -CollectionId <String> -IncludeCollectionId <String> [ <CommonParameters>]
```

Parameter Set: ByCollectionIdAndIncludeCollectionName

```
Get-CMDeviceCollectionIncludeMembershipRule -CollectionId <String> -IncludeCollectionName <String> [ <CommonParameters>]
```

Parameter Set: ByCollectionNameAndIncludeCollectionId

```
Get-CMDeviceCollectionIncludeMembershipRule -CollectionName <String> -IncludeCollectionId <String> [ <CommonParameters>]
```

Parameter Set: ByCollectionValueAndIncludeCollectionId

```
Get-CMDeviceCollectionIncludeMembershipRule -Collection <IResultObject> -IncludeCollectionId <String> [ <CommonParameters>]
```

Parameter Set: ByCollectionValueAndIncludeCollectionName

```
Get-CMDeviceCollectionIncludeMembershipRule -Collection <IResultObject> -IncludeCollectionName <String> [ <CommonParameters>]
```

Detailed Description

The **Get-**CMDeviceCollectionIncludeMembershipRule**** cmdlet retrieves rules that include the members of another collection in the device collections where the rule is applied. You can specify the device collections where the rule is applied by using their names, Ids, or by specifying an object that represents the collections.

Configuration Manager dynamically updates the membership of the device collection if the membership of the included collection changes. For more information about membership rules, see [Introduction to Collections in Configuration Manager](http://go.microsoft.com/fwlink/p/?LinkID=259433) (<http://go.microsoft.com/fwlink/p/?LinkID=259433>) on TechNet.

Parameters

-Collection<IResultObject>

Specifies a Configuration Manager device collection object. To obtain a device collection object, use the **Get-CMDeviceCollection** cmdlet.

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

-CollectionId<String>

Specifies the ID of the device collection where the rule is applied.

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

-CollectionName<String>

Specifies the name of the device collection where the rule is applied.

Aliases	none
Required?	true

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-IncludeCollectionId<String>

Specifies the ID of the collection whose members are included in the device collection.

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

-IncludeCollectionName<String>

Specifies the name of the collection whose members are included in the device collections.

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

<CommonParameters>

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

Examples

Example 1: Get the include membership rules from a device collection

This command gets the include membership rules for the collection that has the ID SMSDM001 from the device collection that has the ID 9990000D.

```
PS C:\> Get-CMDeviceCollectionIncludeMembershipRule -CollectionId "9990000D" -  
IncludeCollectionId "SMSDM001"
```

Related topics

[Add-CMDeviceCollectionIncludeMembershipRule](#)

[Remove-CMDeviceCollectionIncludeMembershipRule](#)

[Get-CMDeviceCollection](#)

Get-CMDeviceCollectionQueryMembershipRule

Get-CMDeviceCollectionQueryMembershipRule

Gets the query membership rules from one or more device collections in the Configuration Manager hierarchy.

Syntax

Parameter Set: ByCollectionName

```
Get-CMDeviceCollectionQueryMembershipRule -CollectionName <String> -RuleName <String> [  
<CommonParameters>]
```

Parameter Set: ByCollectionId

```
Get-CMDeviceCollectionQueryMembershipRule -CollectionId <String> -RuleName <String> [  
<CommonParameters>]
```

Parameter Set: ByCollectionValue

```
Get-CMDeviceCollectionQueryMembershipRule -Collection <IResultObject> -RuleName <String> [  
<CommonParameters>]
```

Detailed Description

The **Get-CMDeviceCollectionQueryMembershipRule** cmdlet retrieves rules from the specified device collections. You can specify the device collections where the rule is applied by using their names, IDs, or by specifying an input object that represents the device collections. The query is specified by its Id or name.

A query rule lets you dynamically update the membership of a collection based on a query that is run on a schedule. For more information about membership rules, see [Introduction to Collections in Configuration Manager](http://go.microsoft.com/fwlink/p/?LinkID=259433) (<http://go.microsoft.com/fwlink/p/?LinkID=259433>) on TechNet.

Parameters

-Collection<IResultObject>

Specifies a Configuration Manager device collection object. To obtain a device collection object, use the **Get-CMDeviceCollection** cmdlet.

Aliases	none
Required?	true

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CollectionId<String>

Specifies the ID of the device collections where the rule is applied.

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

-CollectionName<String>

Specifies the name of the device collection where the rule is applied.

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

-RuleName<String>

Specifies the name of the query rule.

Aliases	none
Required?	true

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

<CommonParameters>

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

Examples

Example 1: Get the query membership rules for a device collection

This command gets the query membership rule named Remote Users By Domain from device collection named Remote Users.

```
PS C:\> Get-CMUserCollectionQueryMembershipRule -CollectionName "Remote Users" -RuleName "Remote Users By Domain"
```

Related topics

[Add-CMUserCollectionQueryMembershipRule](#)

[Remove-CMUserCollectionQueryMembershipRule](#)

[Get-CMUserCollection](#)

[Get-CMDeviceCollection](#)

Get-CMDeviceCollectionVariable

Get-CMDeviceCollectionVariable

Gets the task sequence variables for a device collection in the Configuration Manager hierarchy.

Syntax

Parameter Set: SearchByNameMandatory

```
Get-CMDeviceCollectionVariable -CollectionName <String> -VariableName <String> [  
<CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Get-CMDeviceCollectionVariable -CollectionId <String> -VariableName <String> [  
<CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Get-CMDeviceCollectionVariable -Collection <IResultObject> -VariableName <String> [  
<CommonParameters>]
```

Detailed Description

The **Get-CMDeviceCollectionVariable** cmdlet gets the variables of the task sequences for a device collection.

Task sequence variables are a set of name and value pairs that provide a mechanism to configure and customize the steps of a task sequence when the task sequence is deployed to a specific collection.

For more information about task sequence variables, see [Planning a Task Sequence Strategy in Configuration Manager](http://go.microsoft.com/fwlink/p/?LinkID=260806) (<http://go.microsoft.com/fwlink/p/?LinkID=260806>) on TechNet.

Parameters

-Collection<IResultObject>

Specifies a Configuration Manager user collection object. To obtain a user collection object, use the **Get-CMUserCollection** cmdlet.

Aliases	none
Required?	true
Position?	named

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

-CollectionId<String>

Specifies the ID of the device collection.

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

-CollectionName<String>

Specifies the name of the device collection.

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

-VariableName<String>

Specifies the name of the variable. If you do not specify this parameter, the cmdlet returns all the variables for the collection.

Aliases	none
Required?	true

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

<CommonParameters>

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

Examples

Example 1: Get task sequence variables for all device collections

This command gets the task sequence variables for all device collections.

```
PS C:\> Get-CMDeviceCollectionVariable -CollectionName "" -VariableName ""
```

Related topics

[Set-CMDeviceCollectionVariable](#)

[New-CMDeviceCollectionVariable](#)

[Remove-CMDeviceCollectionVariable](#)

[Get-CMUserCollection](#)

Get-CMDeviceVariable

Get-CMDeviceVariable

Gets device variables.

Syntax

Parameter Set: SearchByIdMandatory

```
Get-CMDeviceVariable -ResourceId <String> -VariableName <String> [ <CommonParameters>]
```

Parameter Set: SearchByNameMandatory

```
Get-CMDeviceVariable -DeviceName <String> -VariableName <String> [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Get-CMDeviceVariable -Device <IResultObject> -VariableName <String> [ <CommonParameters>]
```

Detailed Description

The **Get-CMDeviceVariable** cmdlet gets device variables. Individual devices have device variables. Task sequence processing uses device variables.

Parameters

-Device<IResultObject>

Specifies a **CMDevice** object. To obtain a **CMDevice** object, use the **Get-CMDevice** cmdlet.

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

-DeviceName<String>

Specifies a device name. You can specify a NetBIOS name or a fully qualified domain name (FQDN).

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

-ResourceId<String>

Specifies a Systems Management Server (SMS) ID.

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

-VariableName<String>

Specifies the name of the device variable.

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

<CommonParameters>

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

Examples

Example 1: Get a variable value by using its name

This command gets the value of the variable named HostDrive for the specified computer.

```
PS C:\> Get-CMDeviceVariable -DeviceName "Computer073" -VariableName "HostDrive"
```

Related topics

[New-CMDeviceVariable](#)

[Remove-CMDeviceVariable](#)

[Set-CMDeviceVariable](#)

[Get-CMDevice](#)

Get-CMDiscoveryMethod

Get-CMDiscoveryMethod

Gets a discovery method for Configuration Manager.

Syntax

Parameter Set: SearchDiscoveryMethodsByName

```
Get-CMDiscoveryMethod [-Name {ActiveDirectoryForestDiscovery | ActiveDirectoryGroupDiscovery | ActiveDirectorySystemDiscovery | ActiveDirectoryUserDiscovery | HeartbeatDiscovery | NetworkDiscovery} ] [-SiteCode <String> ] [ <CommonParameters>]
```

Detailed Description

The **Get-CMDiscoveryMethod** cmdlet gets a discovery method for Microsoft System Center 2012 Configuration Manager. Discovery identifies computer and user resources that System Center 2012 Configuration Manager can manage. If it discovers a resource, Configuration Manager creates a record in the System Center 2012 Configuration Manager database for the resource and its associated information. You can then use the discovery information to help you to install the System Center 2012 Configuration Manager client and create custom queries and collections to logically group resources for related management tasks.

For more information about discovery in System Center 2012 Configuration Manager, see [About Configuration Manager Discovery](http://go.microsoft.com/fwlink/?linkid=107444) (<http://go.microsoft.com/fwlink/?linkid=107444>) on TechNet.

Parameters

-Name<DiscoveryType>

Specifies the type of discovery method that the cmdlet gets. Valid values are:

-- ActiveDirectoryForestDiscovery: Discovers security groups, including local, global, and universal groups from specified locations in Active Directory Domain Services.

-- ActiveDirectoryGroupDiscovery: Discovers additional information, including the OU and group membership of the computer, about previously discovered computers from specified locations in Active Directory Domain Services.

-- ActiveDirectorySystemDiscovery: Discovers computers from specified locations in Active Directory Domain Services.

-- ActiveDirectoryUserDiscovery: Discovers users from specified locations in Active Directory Domain Services.

-- HeartbeatDiscovery: Updates discovery records for Microsoft System Center 2012 Configuration Manager clients in the System Center 2012 Configuration Manager database without discovering new resources.

-- NetworkForestDiscovery: Searches the network infrastructure for network devices (such as printers, routers, and bridges) that have an IP address.

The acceptable values for this parameter are:

ActiveDirectoryForestDiscovery	
ActiveDirectoryGroupDiscovery	
ActiveDirectorySystemDiscovery	
ActiveDirectoryUserDiscovery	
HeartbeatDiscovery	
NetworkDiscovery	

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

-SiteCode<String>

Specifies a site code.

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

<CommonParameters>

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

Examples

Example 1: Get a user discovery method

This command gets a System Center 2012 Configuration Manager method that discovers users in the installation.

```
PS C:\> Get-CMDiscoveryMethod -Name "ActiveDirectoryUserDiscovery"
```

Related topics

[Set-CMDiscoveryMethod](#)

Get-CMDistributionPoint

Get-CMDistributionPoint

Gets a distribution point.

Syntax

Parameter Set: SearchByName

```
Get-CMDistributionPoint [-SiteCode <String> ] [-SiteSystemServerName <String> ] [ <CommonParameters>]
```

Parameter Set: SearchByGroup

```
Get-CMDistributionPoint -DistributionPointGroup <IResultObject> [ <CommonParameters>]
```

Parameter Set: SearchByGroupId

```
Get-CMDistributionPoint -DistributionPointGroupId <String> [ <CommonParameters>]
```

Parameter Set: SearchByGroupName

```
Get-CMDistributionPoint -DistributionPointGroupName <String> [ <CommonParameters>]
```

Detailed Description

The **Get-CMDistributionPoint** cmdlet gets a distribution point. Microsoft System Center 2012 Configuration Manager uses distribution points to store files for clients to download, such as application content, software packages, software updates, operating system images, and boot images.

Parameters

-DistributionPointGroup<IResultObject>

Specifies a **CMDistributionPointGroup** object. To obtain a **CMDistributionPointGroup** object, use the **Get-CMDistributionPointGroup** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none

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

-DistributionPointGroupId<String>

Specifies the ID of a distribution point group.

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

-DistributionPointGroupName<String>

Specifies the name of a distribution point group.

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

-SiteCode<String>

Specifies the site code of the site that is associated with a distribution point. The default value is "".

Aliases	none
Required?	false
Position?	named
Default Value	none

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

-SiteSystemServerName<String>

Specifies a fully qualified domain name (FQDN) of the server that hosts the site system role.

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

<CommonParameters>

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

Examples

Example 1: Get a distribution point

This command gets a distribution point that is associated with the site system named CMCEN-DIST02.TSQA.CONTOSCO.COM and the site code CM1.

```
PS C:\> Get-CMDistributionPoint -SiteSystemServerName "CMCEN-DIST02.TSQA.CONTOSCO.COM" -
SiteCode "CM1"
```

Related topics

[Add-CMDistributionPoint](#)

[Remove-CMDistributionPoint](#)

[Set-CMDistributionPoint](#)

[Update-CMDistributionPoint](#)

[Get-CMDistributionPointGroup](#)

Get-CMDistributionPointGroup

Get-CMDistributionPointGroup

Gets distribution point groups.

Syntax

Parameter Set: SearchByName

```
Get-CMDistributionPointGroup [-Name <String> ] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Get-CMDistributionPointGroup -Id <String[]> [ <CommonParameters>]
```

Detailed Description

The **Get-CMDistributionPointGroup** cmdlet gets one or more distribution point groups.

Parameters

-Id<String[]>

Specifies an array of IDs of distribution point groups.

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

-Name<String>

Specifies the name of a distribution point group.

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

<CommonParameters>

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

Examples

Example 1: Get a distribution point group by using an ID

This command get the distribution point group that has the ID 6617708D-0F98-4898-8D05-9E882C23DCB2.

```
PS C:\> Get-CMDistributionPointGroup -Id "{6617708D-0F98-4898-8D05-9E882C23DCB2}"
```

Example 2: Get a distribution point group by using a name

This command gets the distribution point group named Dpg01 and that has the ID FA921CF2-89C9-407D-A21D-FE6947F2C00A.

```
PS C:\> Get-CMDistributionPointGroup -Name "Dpg01" -Id "{FA921CF2-89C9-407D-A21D-FE6947F2C00A}"
```

Related topics

[Set-CMDistributionPointGroup](#)

[New-CMDistributionPointGroup](#)

[Remove-CMDistributionPointGroup](#)

Get-CMDriver

Get-CMDriver

Gets a software driver or a device driver.

Syntax

Parameter Set: SearchByName

```
Get-CMDriver [-Name <String[]> ] [-SecuredScopeNames <String> ] [ <CommonParameters>]
```

Parameter Set: SearchByDriverPackage

```
Get-CMDriver -DriverPackage <IResultObject> [ <CommonParameters>]
```

Parameter Set: SearchByDriverPackageIdMandatory

```
Get-CMDriver -DriverPackageId <String> [ <CommonParameters>]
```

Parameter Set: SearchByDriverPackageNameMandatory

```
Get-CMDriver -DriverPackageName <String> [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Get-CMDriver -Id <String[]> [-SecuredScopeNames <String> ] [ <CommonParameters>]
```

Detailed Description

The **Get-CMDriver** cmdlet gets a software driver or device.

Parameters

-DriverPackage<IResultObject>

Specifies a driver object. To obtain a driver object, use the **Get-CMDriverPackage** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

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

-DriverPackageId<String>

Specifies the identifier for a driver package that uses a particular driver. You can get the identifier by using **Get-CMDriverPackage**.

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

-DriverPackageName<String>

Specifies the name for a driver package that uses a particular driver. You can get the identifier by using **Get-CMDriverPackage**.

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

-Id<String[]>

Specifies an array of identifiers for a driver.

Aliases	CIId
Required?	true
Position?	named

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

-Name<String[]>

Specifies an array of names for a driver.

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

-SecuredScopeNames<String>

Specifies the names of security scopes. A security scope name can be Default or the name of a custom-created security scope.

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

<CommonParameters>

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

Examples

Example 1: Get a device driver

This command gets a device driver that is specified by its inclusion in a driver package.

```
PS C:\> Get-CMDriver -DriverPackageName "Adaptec Embedded SCSI Host"
```

Related topics

[Disable-CMDriver](#)

[Enable-CMDriver](#)

[Import-CMDriver](#)

[Remove-CMDriver](#)

[Set-CMDriver](#)

[Get-CMDriverPackage](#)

Get-CMDriverPackage

Get-CMDriverPackage

Gets a driver package.

Syntax

Parameter Set: SearchByName

```
Get-CMDriverPackage [-Name <String> ] [-SecuredScopeNames <String> ] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Get-CMDriverPackage -Id <String[]> [-SecuredScopeNames <String> ] [ <CommonParameters>]
```

Detailed Description

The **Get-CMDriverPackage** cmdlet gets a driver package.

Parameters

-Id<String[]>

Specifies an array of identifiers for a driver package.

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

-Name<String>

Specifies the name of a driver package.

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

-SecuredScopeNames<String>

Specifies the names of security scopes. A security scope name can be Default or the name of a custom-created security scope.

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

<CommonParameters>

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

Examples

Example 1: Get a driver package that is specified by its identifier.

This command gets a driver package that is specified by its identifier.

```
PS C:\> Get-CMDriverPackage -Id "CM100042"
```

Related topics

[Export-CMDriverPackage](#)

[Import-CMDriverPackage](#)

[New-CMDriverPackage](#)

[Remove-CMDriverPackage](#)

[Set-CMDriverPackage](#)

Get-CMEmailNotificationComponent

Get-CMEmailNotificationComponent

Gets an email notification components.

Syntax

Parameter Set: SearchByNameMandatory

```
Get-CMEmailNotificationComponent -SiteSystemServerName <String[]> [ <CommonParameters>]
```

Parameter Set: SearchBySiteCodeMandatory

```
Get-CMEmailNotificationComponent -SiteCode <String> [-SiteSystemServerName <String[]> ] [ <CommonParameters>]
```

Detailed Description

The **Get-CMEmailNotificationComponent** cmdlet gets one or more email notification components Microsoft System Center 2012 Configuration Manager.

Parameters

-SiteCode<String>

Specifies the three-letter site code of the Configuration Manager site that hosts the site system role.

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

-SiteSystemServerName<String[]>

Specifies an array of fully qualified domain names (FQDN) of the servers that host the site system role.

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

<CommonParameters>

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

Examples

Example 1: Get an email notification component by using a site code

This command gets a notification component for the site that has the site code CM2.

```
PS C:\> Get-CMEmailNotificationComponent -SiteCode "CM2"
```

Example 2: Get an email notification component by using a site system server name

This command gets a notification component for the site that has the server that has the specified name.

```
PS C:\> Get-CMEmailNotificationComponent -SiteSystemServerName "CMCEN-DIST02.TSQA.CORP.CONTOSCO.COM"
```

Related topics

[Set-CMEmailNotificationComponent](#)

Get-CMEndpointProtectionPoint

Get-CMEndpointProtectionPoint

Gets an Endpoint Protection point.

Syntax

Parameter Set: SearchByName

```
Get-CMEndpointProtectionPoint [-SiteCode <String> ] [-SiteSystemServerName <String> ] [ <CommonParameters> ]
```

Detailed Description

The **Get-CMEndpointProtectionPoint** cmdlet gets a System Center 2012 Endpoint Protection point in Microsoft System Center 2012 Configuration Manager. For more information about Endpoint Protection in Microsoft System Center 2012 Configuration Manager, see [Endpoint Protection in Configuration Manager](http://go.microsoft.com/fwlink/?LinkId=268427) (<http://go.microsoft.com/fwlink/?LinkId=268427>) on TechNet.

Parameters

-SiteCode<String>

Specifies a site code.

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

-SiteSystemServerName<String>

Specifies the fully qualified domain name (FQDN) of the server that hosts the site system role.

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

<CommonParameters>

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

Examples

Example 1: Get an Endpoint Protection point

This command gets an Endpoint Protection point.

```
PS C:\> Get-CMEndpointProtectionPoint -SiteCode "CM1" -SiteSystemServerName  
"CMServer01.Contoso.com"
```

Related topics

[Add-CMEndpointProtectionPoint](#)

[Remove-CMEndpointProtectionPoint](#)

[Set-CMEndpointProtectionPoint](#)

Get- CMEndpointProtectionSummarizationSchedule

Get-**CMEndpointProtectionSummarizationSchedule**

Gets an Endpoint Protection summarization schedule.

Syntax

```
Get-CMEndpointProtectionSummarizationSchedule [ <CommonParameters>]
```

Detailed Description

The **Get-**CMEndpointProtectionSummarizationSchedule**** cmdlet gets a System Center 2012 Endpoint Protection summarization schedule. For more information about Endpoint Protection summarization schedules, see [How to Monitor Endpoint Protection in Configuration Manager](http://go.microsoft.com/fwlink/?LinkId=268428) (<http://go.microsoft.com/fwlink/?LinkId=268428>) on TechNet.

Parameters

<CommonParameters>

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

Examples

Example 1: Get an Endpoint Protection summarization schedule

This command gets an Endpoint Protection summarization schedule.

```
PS C:\> Get-CMEndpointProtectionSummarizationSchedule
```

Related topics

[Set-**CMEndpointProtectionSummarizationSchedule**](#)

Get-CMEnrollmentPoint

Get-CMEnrollmentPoint

Gets an enrollment point.

Syntax

Parameter Set: SearchByName

```
Get-CMEnrollmentPoint [-SiteCode <String> ] [-SiteSystemServerName <String> ] [ <CommonParameters> ]
```

Detailed Description

The **Get-CMEnrollmentPoint** cmdlet gets an enrollment point in Microsoft System Center 2012 Configuration Manager. An enrollment point is a site system role that uses public key infrastructure (PKI) certificates to complete mobile device enrollment and to provision Intel AMT-based computers.

Parameters

-SiteCode<String>

Specifies a site code.

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

-SiteSystemServerName<String>

Specifies a fully qualified domain name (FQDN) of the server that hosts the site system role.

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

<CommonParameters>

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

Examples

Example 1: Get an enrollment point

This command gets an enrollment point.

```
PS C:\> Get-CMEnrollmentPoint -SiteSystemServerName "SiteServer01.Contoso.com" -SiteCode "CM1"
```

Related topics

[Add-CMEnrollmentPoint](#)

[Remove-CMEnrollmentPoint](#)

[Set-CMEnrollmentPoint](#)

Get-CMEnrollmentProxyPoint

Get-CMEnrollmentProxyPoint

Gets an enrollment proxy point.

Syntax

Parameter Set: SearchByName

```
Get-CMEnrollmentProxyPoint [-SiteCode <String> ] [-SiteSystemServerName <String> ] [ <CommonParameters> ]
```

Detailed Description

The **Get-CMEnrollmentProxyPoint** cmdlet gets a Microsoft System Center 2012 Configuration Manager enrollment proxy point. An enrollment proxy point is a site system role that manages enrollment requests from mobile devices.

Parameters

-SiteCode<String>

Specifies a site code.

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

-SiteSystemServerName<String>

Specifies the fully qualified domain name (FQDN) of the server that hosts the site system role.

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

<CommonParameters>

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

Examples

Example 1: Get an enrollment proxy point

This command gets an enrollment proxy point.

```
PS C:\> Get-CMEnrollmentProxyPoint -SiteCode "CM1" -SiteSystemServerName  
"SiteServer01.Contoso.com"
```

Related topics

[Add-CMEnrollmentProxyPoint](#)

[Remove-CMEnrollmentProxyPoint](#)

Get-CMExchangeServer

Get-CMExchangeServer

Gets a Configuration Manager Exchange Server object.

Syntax

Parameter Set: SearchBySiteCode

```
Get-CMExchangeServer [-Address <String> ] [-SiteCode <String> ] [ <CommonParameters>]
```

Detailed Description

The **Get-CMExchangeServer** cmdlet gets an object that represents a Microsoft Exchange Server that Microsoft System Center 2012 Configuration Manager uses.

System Center 2012 Configuration Manager works with Exchange Server to manage mobile devices that cannot run System Center 2012 Configuration Manager clients.

Parameters

-Address<String>

Specifies a URL for the Exchange Server.

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

-SiteCode<String>

Specifies the site code for a Configuration Manager site associated with the Exchange Server.

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

<CommonParameters>

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

Examples

Example 1: Get all Exchange Server systems

This command gets all the Exchange Server items for a System Center 2012 Configuration Manager server.

```
PS C:\> Get-CMExchangeServer
```

Example 2: Get an Exchange Server for a site

This command gets an Exchange Server for the site identified by the site code PE1.

```
PS C:\> Get-CMExchangeServer -SiteCode "PE1"
```

Example 3: Get a specified Exchange Server

This command gets the Exchange Server with the specified address.

```
PS C:\> Get-CMExchangeServer -Address "http://localhost/PowerShell"
```

Related topics

[New-CMExchangeServer](#)

[Remove-CMExchangeServer](#)

[Set-CMExchangeServer](#)

[Sync-CMExchangeServer](#)

Get-CMFallbackStatusPoint

Get-CMFallbackStatusPoint

Gets a Configuration Manager fallback status point.

Syntax

Parameter Set: SearchByName

```
Get-CMFallbackStatusPoint [-SiteCode <String> ] [-SiteSystemServerName <String> ] [ <CommonParameters> ]
```

Detailed Description

The **Get-CMFallbackStatusPoint** cmdlet gets a fallback status point site server role. You can get fallback status point for a site system name or a site code or both.

Microsoft System Center 2012 Configuration Manager can use one or more fallback status points to collect state messages for a site and send them on to System Center 2012 Configuration Manager. You can use this cmdlet to get a fallback status point to use with other cmdlets, such as the **Set-CMFallbackStatusPoint** cmdlet or the **Remove-CMFallBackStatusPoint** cmdlet.

Parameters

-SiteCode<String>

Specifies the site code for a fallback status point.

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

-SiteSystemServerName<String>

Specifies the site system name for a fallback status point.

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

<CommonParameters>

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

Examples

Example 1: Get a fallback status point

This command gets a fallback status point for the site with the site code cm1 and the system name Server21.West01.Contoso.com. You can have more than one fallback status point for a site. This example specifies both the site code and the server name.

```
PS C:\> Get-CMFallbackStatusPoint -SiteCode "CM1" -SiteSystemServerName  
"Server21.West01.Contoso.com"
```

Related topics

[Add-CMFallbackStatusPoint](#)

[Remove-CMFallbackStatusPoint](#)

[Set-CMFallbackStatusPoint](#)

Get-CMFileReplicationRoute

Get-CMFileReplicationRoute

Gets a file replication route from Configuration Manager.

Syntax

Parameter Set: SearchBySiteCode

```
Get-CMFileReplicationRoute [-DestinationSiteCode <String> ] [-SourceSiteCode <String> ] [ <CommonParameters> ]
```

Parameter Set: SearchBySiteName

```
Get-CMFileReplicationRoute [-DestinationSiteName <String> ] [-SourceSiteName <String> ] [ <CommonParameters> ]
```

Detailed Description

The **Get-CMFileReplicationRoute** cmdlet gets a file replication route from Microsoft System Center 2012 Configuration Manager. System Center 2012 Configuration Manager uses file replication routes to transfer file-based data between sites in a hierarchy. Each file replication route identifies a destination site to which file-based data can transfer.

File replication were known as addresses in versions of Configuration Manager before System Center 2012 Configuration Manager. The functionality of file replication routes is the same as that of addresses in earlier versions.

Parameters

-DestinationSiteCode<String>

Specifies a destination site for data transfers by using a site code.

Aliases	DesSiteCode
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

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

-DestinationSiteName<String>

Specifies a destination site for data transfers by using a site name.

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

-SourceSiteCode<String>

Specifies a source site for data transfers by using a site code.

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

-SourceSiteName<String>

Specifies a destination site for data transfers by using a site name.

Aliases	SiteName
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?

false

<CommonParameters>

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

Examples

Example 1: Get a file replication route by using site codes

This command creates a file replication route from the site that has the site code IM1 to the site that has the site code IM5.

```
PS C:\> Get-CMFileReplicationRoute -DestinationSiteCode "IM5" -SourceSiteCode "IM1"
```

Related topics

[New-CMFileReplicationRoute](#)

[Remove-CMFileReplicationRoute](#)

[Set-CMFileReplicationRoute](#)

Get-CMGlobalCondition

Get-CMGlobalCondition

Gets global condition objects.

Syntax

Parameter Set: SearchByName

```
Get-CMGlobalCondition [-Name <String[]> ] [-SecuredScopeNames <String> ] [ <CommonParameters> ]
```

Parameter Set: SearchByIdMandatory

```
Get-CMGlobalCondition -Id <String[]> [-SecuredScopeNames <String> ] [ <CommonParameters> ]
```

Detailed Description

The **Get-CMGlobalCondition** cmdlet gets global condition objects. You can pass the results of this cmdlet to the **Set-CMGlobalCondition** cmdlet or the **Remove-CMGlobalCondition** cmdlet.

Microsoft System Center 2012 Configuration Manager uses global conditions to represent business or technical conditions. Global conditions specify how to provide and deploy applications to client devices.

You can get global conditions by name, ID, or security scope. You can also specify one or more security scope names with either names or IDs. For instance, you might specify an array of global condition names and specify a security scope to narrow your results.

Parameters

-Id<String[]>

Specifies an array of identifiers of global conditions. This value corresponds to the **CI_ID** property of a global condition object.

Aliases	CIId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

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

-Name<String[]>

Specifies an array of names for global conditions. This value corresponds to the **LocalizedDisplayName** property of a global condition object.

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

-SecuredScopeNames<String>

Specifies a security scope of a global condition. You can use this parameter to narrow your specified global conditions by the current security scope.

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

<CommonParameters>

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

Examples

Example 1: Get a global condition by name

This command gets the global condition named CPU speed.

```
PS C:\> Get-CMGlobalCondition -Name "CPU speed"
```

Example 2: Get a global condition by name and security scope

This command gets the global condition named CPU speed that has a security scope named Scope22.

```
PS C:\> Get-CMGlobalCondition -Name "CPU speed" -SecuredScopeNames "Scope22"
```

Related topics

[New-CMGlobalCondition](#)

[Remove-CMGlobalCondition](#)

[Set-CMGlobalCondition](#)

Get-CMHardwareRequirement

Get-CMHardwareRequirement

Gets Configuration Manager hardware requirements for products.

Syntax

Parameter Set: SearchByName

```
Get-CMHardwareRequirement [-Product <String> ] [ <CommonParameters>]
```

Detailed Description

The **Get-CMHardwareRequirement** cmdlet gets hardware requirements objects for software products.

Microsoft System Center 2012 Configuration Manager manages Asset Intelligence information, including hardware requirements, for different software products. You can add, modify, or delete your own hardware requirements, but you cannot change built-in hardware requirement objects.

You can use this cmdlet to get all the hardware requirement objects for a System Center 2012 Configuration Manager server or one or more hardware requirement objects for a specified product names. You can use hardware requirements with other cmdlets, such as the **Remove-CMHardwareRequirement** cmdlet or the **Set-CMHardwareRequirement** cmdlet.

Parameters

-Product<String>

Specifies of the name of a software product.

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

<CommonParameters>

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

Examples

Example 1: Get a hardware requirement

This command gets the hardware requirement object for a product named Accounts Program.

```
PS C:\> Get-CMHardwareRequirement -Product "Accounts Program"
```

```
IsLocal      : False
MinCPU       : 233
MinDiskFree  : 1572864
MinDiskSize  : 10485760
MinRAM       : 131072
Product      : Accounts Program
State        : 0
```

Related topics

[New-CMHardwareRequirement](#)

[Remove-CMHardwareRequirement](#)

[Set-CMHardwareRequirement](#)

Get-CMInitModifiableSecuredCategory

Get-CMInitModifiableSecuredCategory

Syntax

Parameter Set: SearchById

```
Get-CMInitModifiableSecuredCategory [-Id <String> ] [-ObjectTypeId <String> ] [ <CommonParameters> ]
```

Parameter Set: SearchByName

```
Get-CMInitModifiableSecuredCategory [-Name <String> ] [-ObjectTypeId <String> ] [ <CommonParameters> ]
```

Detailed Description

The **Get-CMInitialModifiableSecuredCategory** cmdlet

Parameters

-Id<String>

Specifies an identifier in Configuration Manager.

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

-Name<String>

Specifies a name in Configuration Manager.

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

-ObjectTypeId<String>

Specifies an ID for an object type.

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

<CommonParameters>

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

Get-CMIPSubnet

Get-CMIPSubnet

Gets a Configuration Manager IP subnet.

Syntax

Parameter Set: SearchByName

```
Get-CMIPSubnet [-Name <String[]> ] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Get-CMIPSubnet -Id <String[]> [ <CommonParameters>]
```

Detailed Description

The **Get-CMIPSubnet** cmdlet gets an IP subnet object that Microsoft System Center 2012 Configuration Manager uses as a boundary.

A boundary is a network location on the intranet that can contain one or more devices that you want to manage. System Center 2012 Configuration Manager can define a boundary in several ways, including an IP subnet. For more information about boundaries, see [Planning for Boundaries and Boundary Groups in Configuration Manager](http://go.microsoft.com/fwlink/?LinkId=268431) (<http://go.microsoft.com/fwlink/?LinkId=268431>) on TechNet.

Parameters

-Id<String[]>

Specifies an array of IDs for IP subnets. This is a Configuration Manager name, not an IP address or IP address and subnet.

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

-Name<String[]>

Specifies an array of names for IP subnets.

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

<CommonParameters>

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

Examples

Example 1: Get an IP subnet

This command gets the IP subnet object named West07Subnet.

```
PS C:\> Get-CMIPSubnet -Name "West07Subnet"
```

Related topics

[Get-CMActiveDirectorySite](#)

Get-CMMaintenanceWindow

Get-CMMaintenanceWindow

Gets the maintenance windows for a collection.

Syntax

Parameter Set: Default

```
Get-CMMaintenanceWindow [-CollectionID] <String[]> [ <CommonParameters>]
```

Detailed Description

The **Get-CMMaintenanceWindow** cmdlet gets the maintenance windows for specified collections.

Parameters

-CollectionID<String[]>

Specifies an array of collection IDs.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Get maintenance windows

This command gets the maintenance windows for the specified collection.

```
PS C:\> Get-CMMaintenanceWindow -CollectionID "AAA0004D"
```

Related topics

[New-CMMaintenanceWindow](#)

[Remove-CMMaintenanceWindow](#)

[Set-CMMaintenanceWindow](#)

Get-CMManagementPoint

Get-CMManagementPoint

Gets a management point.

Syntax

Parameter Set: SearchByName

```
Get-CMManagementPoint [-SiteCode <String> ] [-SiteSystemServerName <String> ] [ <CommonParameters> ]
```

Detailed Description

The **Get-CMManagementPoint** cmdlet gets a management point. A management point is a site system role that provides policy and service location information to clients and receives configuration data from clients.

Parameters

-SiteCode<String>

Specifies the site code of the Configuration Manager site that hosts the site system role.

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

-SiteSystemServerName<String>

Specifies the name of the server that hosts the site system role.

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

<CommonParameters>

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

Examples

Example 1: Get a management point

This command gets a management point that is associated with the site system named cmcendist02.tsqa.contoso.com and the site code CM1. The command directs the output to the file Get-CMOutOfBandServicePoint_data.txt.

```
PS C:\> Get-CMManagementPoint -SiteSystemServerName "cmcendist02.tsqa.contoso.com" -SiteCode "CM1" >>\CMmgmt01\Get-CMManagementPoint_data.txt
```

Related topics

[Add-CMManagementPoint](#)

[Remove-CMManagementPoint](#)

[Get-CMManagementPointComponent](#)

Get-CMManagementPointComponent

Get-CMManagementPointComponent

Gets a component for a Configuration Manager management point.

Syntax

Parameter Set: SearchByNameMandatory

```
Get-CMManagementPointComponent -SiteCode <String> [ <CommonParameters>]
```

Parameter Set: SearchBySiteCodeMandatory

```
Get-CMManagementPointComponent -SiteSystemServerName <String> [ <CommonParameters>]
```

Detailed Description

The **Get-CMManagementPointComponent** cmdlet gets a component of a management point for Microsoft System Center 2012 Configuration Manager. A management point is a System Center 2012 Configuration Manager site that provides policy and service information to clients and receives configuration data from clients.

Parameters

-SiteCode<String>

Specifies the site code for the management point.

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

-SiteSystemServerName<String>

Specifies a fully qualified domain name (FQDN) of the server that hosts the site system role.

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

<CommonParameters>

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

Examples

Example 1: Get a management point component

This command gets a component that is associated with the site that has the code CM1. The command directs the output to the file Get-CMManagementPointComponent_data.txt.

```
PS C:\> Get-CMManagementPointComponent -SiteCode "CM1" >>\1\Get-CMManagementPointComponent_data.txt
```

Related topics

[Set-CMManagementPointComponent](#)

[Get-CMManagementPoint](#)

[Remove-CMManagementPoint](#)

Get-CMMigrationCollection

Get-CMMigrationCollection

Gets collections selected for migration.

Syntax

Parameter Set: SearchById

```
Get-CMMigrationCollection [-Id <String> ] [ <CommonParameters>]
```

Parameter Set: SearchByName

```
Get-CMMigrationCollection [-Name <String> ] [ <CommonParameters>]
```

Detailed Description

The **Get-CMMigrationCollection** cmdlet gets the collections selected from a source hierarchy for migration. A collection is a set of resources in the Microsoft System Center 2012 Configuration Manager hierarchy. A migration collection is the set of resources chosen from a hierarchy for migration, including related objects.

Parameters

-Id<String>

Specifies an identifier for a collection in Configuration Manager.

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

-Name<String>

Specifies a name for a collection in Configuration Manager.

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

<CommonParameters>

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

Examples

Example 1: Get a migration collection by name

This command uses the **Get-CMMigrationCollection** cmdlet to get the migration collection. The command specifies the value PhoneCollection5 for the *Name* parameter.

```
PS C:\> Get-CMMigrationCollection -Name "PhoneCollection5"
```

Get-CMMigrationEntity

Get-CMMigrationEntity

Gets a migration entity in System Center 2012 Configuration Manager.

Syntax

Parameter Set: SearchById

```
Get-CMMigrationEntity [-Id <String> ] [ <CommonParameters>]
```

Parameter Set: SearchByName

```
Get-CMMigrationEntity [-Name <String> ] [ <CommonParameters>]
```

Parameter Set: SearchByOthers

```
Get-CMMigrationEntity [-IsActive <String> ] [-Type <String> ] [ <CommonParameters>]
```

Detailed Description

The **Get-CMMigrationEntity** cmdlet gets the migration entity in Microsoft System Center 2012 Configuration Manager. A migration entity is an object to be migrated that is of any type that is supported by migration.

Parameters

-Id<String>

Specifies an identifier in Configuration Manager.

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

-IsActive<String>

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

-Name<String>

Specifies a name in Configuration Manager.

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

-Type<String>

Specifies a type in Configuration Manager.

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

<CommonParameters>

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

Related topics

[Get-CMMigrationEntityDependency](#)

[New-CMMigrationJob](#)

Get-CMMigrationEntityDependency

Get-CMMigrationEntityDependency

Gets a dependency for a migration entity in System Center 2012 Configuration Manager.

Syntax

Parameter Set: SearchById

```
Get-CMMigrationEntityDependency [-Id <String> ] [ <CommonParameters>]
```

Parameter Set: SearchByType

```
Get-CMMigrationEntityDependency [-Type <String> ] [ <CommonParameters>]
```

Detailed Description

The **Get-CMMigrationEntityDependency** cmdlet gets a migration entity dependency in Microsoft System Center 2012 Configuration Manager. A migration entity dependency is an object upon which another object to be migrated is dependent.

Parameters

-Id<String>

Specifies an identifier in Configuration Manager.

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

-Type<String>

Specifies a type in Configuration Manager.

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

<CommonParameters>

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

Get-CMOperatingSystemImage

Get-CMOperatingSystemImage

Gets operating system images.

Syntax

Parameter Set: SearchByName

```
Get-CMOperatingSystemImage [-Name <String> ] [-SecuredScopeNames <String> ] [ <CommonParameters> ]
```

Parameter Set: SearchByIdMandatory

```
Get-CMOperatingSystemImage -Id <String[]> [-SecuredScopeNames <String> ] [ <CommonParameters> ]
```

Detailed Description

The **Get-CMOperatingSystemImage** cmdlet gets one or more operating system images on a Microsoft System Center 2012 Configuration Manager site. Operating system images are .wim format files and represent a compressed collection of reference files and folders that System Center 2012 Configuration Manager requires to successfully install and configure an operating system on a computer.

Parameters

-Id<String[]>

Specifies an array of IDs of operating system images.

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

-Name<String>

Specifies the name of an operating system image.

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

-SecuredScopeNames<String>

Specifies names of security scopes. A security scope name can be Default or the name of a custom-created security scope.

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

<CommonParameters>

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

Examples

Example 1: Get an operating system image

This command gets the operating system image named OSImagePkg01 for the security scope named SecScope02.

```
PS C:\> Get-CMOperatingSystemImage -Name "OSImagePkg01" -SecuredScopeNames "SecScope02"
```

Related topics

[New-CMOperatingSystemImage](#)

[Set-CMOperatingSystemImage](#)

[Remove-CMOperatingSystemImage](#)

[Get-CMOperatingSystemImageUpdateSchedule](#)

Get-CMOperatingSystemImageUpdateSchedule

Get-CMOperatingSystemImageUpdateSchedule

Retrieves an operating system image update schedule object in Configuration Manager.

Syntax

Parameter Set: SearchByNameMandatory

```
Get-CMOperatingSystemImageUpdateSchedule -Name <String> [-SecuredScopeNames <String> ] [ <CommonParameters> ]
```

Parameter Set: SearchByIdMandatory

```
Get-CMOperatingSystemImageUpdateSchedule -Id <String[]> [-SecuredScopeNames <String> ] [ <CommonParameters> ]
```

Parameter Set: SearchByValueMandatory

```
Get-CMOperatingSystemImageUpdateSchedule -InputObject <IResultObject> [ <CommonParameters> ]
```

Detailed Description

The **Get-CMOperatingSystemImageUpdateSchedule** cmdlet retrieves an object that represents an operating system image update schedule in Microsoft System Center 2012 Configuration Manager.

Parameters

-Id<String[]>

Specifies an array of identifiers of operating system image update schedules in Configuration Manager.

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

-InputObject<IResultObject>

Specifies an operating system image update schedule object in Configuration Manager.

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

-Name<String>

Specifies a name of an operating system image update schedule in Configuration Manager.

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

-SecuredScopeNames<String>

Specifies a name of a security scope.

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

<CommonParameters>

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

Examples

Example 1: Retrieve the operating system image update schedule

This command retrieves the operating system image update schedule identified by the ID 1207.

```
PS C:\> Get-CMOperatingSystemImageUpdateSchedule -Id "1207"
```

Related topics

[Clear-CMOperatingSystemImageUpdateSchedule](#)

[Set-CMOperatingSystemImageUpdateSchedule](#)

Get-CMOperatingSystemInstaller

Get-CMOperatingSystemInstaller

Gets operating system installers.

Syntax

Parameter Set: SearchByName

```
Get-CMOperatingSystemInstaller [-Name <String> ] [-SecuredScopeNames <String> ] [ <CommonParameters> ]
```

Parameter Set: SearchByIdMandatory

```
Get-CMOperatingSystemInstaller -Id <String[]> [-SecuredScopeNames <String> ] [ <CommonParameters> ]
```

Detailed Description

The **Get-CMOperatingSystemInstaller** cmdlet gets one or more operating system installers. An operating system installer is an installation package that contains all the files that Microsoft System Center 2012 Configuration Manager needs to install a Windows operating system on a reference computer.

Parameters

-Id<String[]>

Specifies an array of IDs of operating system installers.

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

-Name<String>

Specifies the name of an operating system installer.

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

-SecuredScopeNames<String>

Specifies names of security scopes. A security scope name can be Default or the name of a custom-created security scope.

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

<CommonParameters>

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

Examples

Example 1: Get an operating system installer

This command gets the operating system installer named OSInstPkg01 for the security scope named SecScope02.

```
PS C:\> Get-CM0peratingSystemInstaller -Name "OSInstPkg01" -SecuredScopeNames "SecScope02"
```

Related topics

[New-CMOperatingSystemInstaller](#)

[Remove-CMOperatingSystemInstaller](#)

[Set-CMOperatingSystemInstaller](#)

Get-CMOutOfBandManagementComponent

Get-CMOutOfBandManagementComponent

Gets an out of band management component.

Syntax

Parameter Set: SearchByNameMandatory

```
Get-CMOutOfBandManagementComponent -SiteSystemServerName <String[]> [ <CommonParameters>]
```

Parameter Set: SearchBySiteCodeMandatory

```
Get-CMOutOfBandManagementComponent -SiteCode <String> [-SiteSystemServerName <String[]> ] [ <CommonParameters>]
```

Detailed Description

The **Get-CMOutOfBandManagementComponent** cmdlet gets the site system computer that has the out of band management role. The out of band management role manages computers that have the Intel vPro chip set and a version of Intel Active Management Technology (Intel AMT) that System Center 2012 Configuration Manager supports. Out of band management lets you connect to a computer AMT management controller when the computer is turned off, in hibernation, or otherwise unresponsive through the operating system.

Parameters

-SiteCode<String>

Specifies assigned site of a client by using a code.

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

-SiteSystemServerName<String[]>

Specifies an array of site system server names.

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

<CommonParameters>

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

Examples

Example 1: Get an out of band management component by using a site code

This command gets the out of band management component from the client site that has code CM4.

```
PS C:\> Get-CMOutOfBandManagementComponent -SiteCode "CM4"
```

Example 2: Get an out of band management component by using a site server name

This command gets the out of band management component from the site server named condev-test04.tsqa.corp.contoso.com in the client site.

```
PS C:\> Get-CMOutOfBandManagementComponent -SiteSystemServerName "condev-test04.tsqa.corp.contoso.com"
```

Related topics

[Set-CMOutOfBandManagementComponent](#)

Get-CMOutOfBandServicePoint

Get-CMOutOfBandServicePoint

Gets an out-of-band service point.

Syntax

Parameter Set: SearchByName

```
Get-CMOutOfBandServicePoint [-SiteCode <String> ] [-SiteSystemServerName <String> ] [ <CommonParameters> ]
```

Detailed Description

The **Get-CMOutOfBandServicePoint** cmdlet gets an out-of-band service point. An out-of-band service point is a site system role that provisions and configures Intel Active Management Technology (AMT)-based computers for Microsoft System Center 2012 Configuration Manager.

Parameters

-SiteCode<String>

Specifies the site code of the Configuration Manager site that hosts the site system role.

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

-SiteSystemServerName<String>

Specifies a fully qualified domain name (FQDN) of the server that hosts the site system role.

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

<CommonParameters>

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

Examples

Example 1: Get an out-of-band service point

This command get an out-of-band service point that is associated with the site system named CMCEN-DIST02.TSQA.CORP.CONTOSCO.COM and the site code CM1. The command directs the output to the text file Get-CMOutOfBandServicePoint_data.txt.

```
PS C:\> Get-CMOutOfBandServicePoint -SiteSystemServerName "cmcen-  
dist02.tsqa.corp.contoso.com" -SiteCode "CM1" >>\Results\Get-  
CMOutOfBandServicePoint_data.txt"
```

Related topics

[Set-CMOutOfBandServicePoint](#)

[Add-CMOutOfBandServicePoint](#)

[Remove-CMOutOfBandServicePoint](#)

Get-CMPackage

Get-CMPackage

Gets Configuration Manager packages.

Syntax

Parameter Set: SearchByName

```
Get-CMPackage [-Name <String> ] [-SecuredScopeNames <String> ] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Get-CMPackage -Id <String[]> [-SecuredScopeNames <String> ] [ <CommonParameters>]
```

Detailed Description

The **Get-CMPackage** cmdlet gets Microsoft System Center 2012 Configuration Manager packages. System Center 2012 Configuration Manager uses packages to distribute software to clients. You can use the *SecuredScopeNames* parameter to specify the security scope of a package to get.

Parameters

-Id<String[]>

Specifies an array of package IDs.

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

-Name<String>

Specifies the name of a package.

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

-SecuredScopeNames<String>

Specifies names of security scopes. A security scope name can be Default or the name of a custom-created security scope.

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

<CommonParameters>

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

Examples

Example 1: Get all packages

This command gets all Configuration Manager packages.

```
PS C:\> Get-CMPackage
```

Example 2: Get a package by using an ID

This command gets the program that has the ID CM100002.

```
PS C:\> Get-CMPackage -Id "CM100002"
```

Example 3: Get a package by using a name

This command gets the program named Configuration Manager Client Package.

```
PS C:\> Get-CMPackage -Name "Configuration Manager Client Package"
```

Related topics

[Export-CMPackage](#)

[Import-CMPackage](#)

[New-CMPackage](#)

[Remove-CMPackage](#)

[Set-CMPackage](#)

Get-CMProgram

Get-CMProgram

Gets programs in Configuration Manager.

Syntax

Parameter Set: SearchByIdAndName

```
Get-CMProgram [-PackageId <String> ] [-ProgramName <String> ] [ <CommonParameters>]
```

Parameter Set: SearchByIdAndNameMandatory

```
Get-CMProgram -PackageId <String> -ProgramName <String> [ <CommonParameters>]
```

Detailed Description

The **Get-CMProgram** cmdlet gets one or more programs in Microsoft System Center 2012 Configuration Manager. Programs are commands that are associated with a System Center 2012 Configuration Manager package. Programs identify the actions that occur when the client receives the client package. You can associate multiple programs with the same package.

Parameters

-PackageId<String>

Specifies the package that contains the program by using an ID.

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

-ProgramName<String>

Specifies the program within the package by using a name.

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

<CommonParameters>

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

Examples

Example 1: Get all programs

This command gets all programs in System Center 2012 Configuration Manager.

```
PS C:\> Get-CMProgram
```

Example 2: Get a program by using a name and an ID

This command gets the program named ProgramD02 in the package that has the ID ST10000F.

```
PS C:\> Get-CMProgram -PackageId "ST10000F" -ProgramName "ProgramD02"
```

Related topics

[Enable-CMProgram](#)

[Disable-CMProgram](#)

[Remove-CMProgram](#)

Get-CMQueryResultMaximum

Get-CMQueryResultMaximum

Gets the maximum number of rows that a Configuration Manager report query can return.

Syntax

```
Get-CMQueryResultMaximum [ <CommonParameters>]
```

Detailed Description

The **Get-CMQueryResultMaximum** cmdlet gets the maximum number of rows that a Microsoft System Center 2012 Configuration Manager report query can return. By default, report queries in System Center 2012 Configuration Manager return only a subset of the matching rows in the database. This cmdlet indicates the number of rows that System Center 2012 Configuration Manager can currently return. You can use the **Set-CMQueryResultMaximum** to change the maximum number of rows that a System Center 2012 Configuration Manager report query can return.

Parameters

<CommonParameters>

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

Examples

Example 1: Get the report query result maximum

This command gets the maximum number of rows that a Configuration Manager report query can return.

```
PS C:\> Get-CMQueryResultMaximum
```

Related topics

[Set-CMQueryResultMaximum](#)



Get- CMRemoteConnectionProfileConfigurationItem

Get-**CMRemoteConnectionProfileConfigurationItem**

Gets remote connection profiles.

Syntax

Parameter Set: SearchByName

```
Get-CMRemoteConnectionProfileConfigurationItem [-Name <String[]> ] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Get-CMRemoteConnectionProfileConfigurationItem -Id <String[]> [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Get-CMRemoteConnectionProfileConfigurationItem -InputObject <IResultObject> [ <CommonParameters>]
```

Detailed Description

The **Get-*CMRemoteConnectionProfileConfigurationItem*** cmdlet gets remote connection profiles.

Parameters

-Id<String[]>

Specifies an array of IDs for remote connection profiles.

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

-InputObject<IResultObject>

Specifies a remote connection profile object.

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

-Name<String[]>

Specifies an array of names of remote connection profiles.

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

<CommonParameters>

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

Examples

Example 1: Get a remote connection profile

This command gets the remote connection profile named Corp_RD_Gateway.

```
PS C:\> Get-CMRemoteConnectionProfileConfigurationItem -Name "Corp_RD_Gateway"
```

Related topics

[Copy-CMRemoteConnectionProfileConfigurationItem](#)

[New-CMRemoteConnectionProfileConfigurationItem](#)

[Remove-CMRemoteConnectionProfileConfigurationItem](#)

[Set-CMRemoteConnectionProfileConfigurationItem](#)

[Get-CMRemoteConnectionProfileConfigurationItemXmlDefinition](#)

Get- CMRemoteConnectionProfileConfigurationItem XmlDefinition

Get-CMRemoteConnectionProfileConfigurationItemXmlDefinition

Gets XML definitions for remote connection profiles.

Syntax

Parameter Set: SearchByNameMandatory

```
Get-CMRemoteConnectionProfileConfigurationItemXmlDefinition -Name <String[]> [  
<CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Get-CMRemoteConnectionProfileConfigurationItemXmlDefinition -Id <String[]> [  
<CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Get-CMRemoteConnectionProfileConfigurationItemXmlDefinition -InputObject <IResultObject> [  
<CommonParameters>]
```

Detailed Description

The **Get-CMRemoteConnectionProfileConfigurationItemXmlDefinition** cmdlet gets XML definitions for remote connection profiles. Client computers use remote connection profiles to remotely connect to computers from outside the domain or over the Internet.

Parameters

-Id<String[]>

Specifies an array of IDs of remote connection profiles.

Aliases	CId
Required?	true
Position?	named

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

-InputObject<IResultObject>

Specifies a remote connection profile object. To obtain a remote connection profile object, use the **Get-CMRemoteConnectionProfileConfigurationItem** cmdlet.

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

-Name<String[]>

Specifies an array of names of remote connection profiles.

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

<CommonParameters>

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

Examples

Example 1: Get a formatted remote connection profile

This command gets a remote connection profile formatted in XML for the item that has the specified identifier.

```
PS C:\> Get-CMRemoteConnectionProfileConfigurationItemXmlDefinition -Id "16777568"
```

Related topics

[Copy-CMRemoteConnectionProfileConfigurationItem](#)

[Get-CMRemoteConnectionProfileConfigurationItem](#)

[New-CMRemoteConnectionProfileConfigurationItem](#)

[Remove-CMRemoteConnectionProfileConfigurationItem](#)

[Set-CMRemoteConnectionProfileConfigurationItem](#)

Get-CMReportingServicePoint

Get-CMReportingServicePoint

Gets a reporting service point.

Syntax

Parameter Set: SearchByName

```
Get-CMReportingServicePoint [-SiteCode <String> ] [-SiteSystemServerName <String> ] [ <CommonParameters> ]
```

Detailed Description

The **Get-CMReportingServicePoint** cmdlet gets a reporting service point. A reporting service point is a site system role that is installed on a server that is running Microsoft SQL Server Reporting Services.

Parameters

-SiteCode<String>

Specifies the site code of the Microsoft System Center 2012 Configuration Manager site that hosts the site system role.

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

-SiteSystemServerName<String>

Specifies a fully qualified domain name (FQDN) of the server that hosts the site system role.

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

<CommonParameters>

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

Examples

Example 1: Get a reporting service point

This command gets a reporting service point from the Configuration Manager site that has the site code CM1 on the site system server named CMCEN-DIST02.TSQA.CORP.CONTOSCO.COM. The command directs the output to the file \Cmrsp01\Get-CMReportingServicePoint_data.txt.

```
PS C:\> Get-CMReportingServicePoint -SiteSystemServerName "CMCEN-  
DIST02.TSQA.CORP.CONTOSCO.COM" -SiteCode "CM1" >>\Cmrsp01\Get-  
CMReportingServicePoint_data.txt
```

Related topics

[Add-CMReportingServicePoint](#)

[Remove-CMReportingServicePoint](#)

[Set-CMReportingServicePoint](#)

Get-CMSecurityRole

Get-CMSecurityRole

Gets security roles.

Syntax

Parameter Set: SearchByName

```
Get-CMSecurityRole [-Name <String[]> ] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Get-CMSecurityRole -Id <String[]> [ <CommonParameters>]
```

Detailed Description

The **Get-CMSecurityRole** cmdlet gets one or more security roles in Microsoft System Center 2012 Configuration Manager.

Parameters

-Id<String[]>

Specifies an array of IDs of security roles.

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

-Name<String[]>

Specifies an array of names of security roles.

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

<CommonParameters>

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

Examples

Example 1: Get all security roles

This command gets all security roles in System Center 2012 Configuration Manager.

```
PS C:\> Get-CMSecurityRole
```

Example 2: Get a security role by using an ID

This command gets the security role that has the ID SMS000CR.

```
PS C:\> Get-CMSecurityRole -Id "SMS000CR"
```

Example 3: Get a security role by using a wildcard

This command gets all security roles that have a name that starts with App.

```
PS C:\> Get-CMSecurityRole -Name App*
```

Related topics

[Set-CMSecurityRole](#)

[Copy-CMSecurityRole](#)

[Export-CMSecurityRole](#)

[Import-CMSecurityRole](#)

[Remove-CMSecurityRole](#)

[Remove-CMSecurityRoleFromAdministrativeUser](#)



Get-CMSecurityScope

Get-CMSecurityScope

Gets security scopes.

Syntax

Parameter Set: SearchByName

```
Get-CMSecurityScope [-Name <String[]> ] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Get-CMSecurityScope -Id <String[]> [ <CommonParameters>]
```

Detailed Description

The **Get-CMSecurityScope** cmdlet gets one or more security scopes in Microsoft System Center 2012 Configuration Manager.

Parameters

-Id<String[]>

Specifies an array of IDs of security scopes.

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

-Name<String[]>

Specifies an array of names of security scopes.

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

<CommonParameters>

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

Examples

Example 1: Get all security scopes

This command gets all the security scopes on the local computer.

```
PS C:\> Get-CMSecurityScope
```

Example 2: Get a security scope by using an ID

This command gets the security scope that has the ID CM100001.

```
PS C:\> Get-CMSecurityScope -Id "CM100001"
```

Example 3: Get security scopes by using a wild card

This command gets security scopes that have a display name that begins with the letter D.

```
PS C:\> Get-CMSecurityScope -Name D*
```

Related topics

[New-CMSecurityScope](#)

[Set-CMSecurityScope](#)

[Remove-CMSecurityScope](#)

[Remove-CMSecurityScopeFromAdministrativeUser](#)

Get-CMSite

Get-CMSite

Gets one or more Configuration Manager sites.

Syntax

Parameter Set: SearchByName

```
Get-CMSite [-Name <String> ] [ <CommonParameters>]
```

Parameter Set: SearchBySiteCodeMandatory

```
Get-CMSite -SiteCode <String> [ <CommonParameters>]
```

Detailed Description

The **Get-CMSite** cmdlet gets one or more Microsoft System Center 2012 Configuration Manager sites. A System Center 2012 Configuration Manager site is a server that has clients assigned to it and that processes client-generated data. You can get a Configuration Manager site by using either a site name or a site code.

Parameters

-Name<String>

Specifies the name of a Configuration Manager site.

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

-SiteCode<String>

Specifies a site code for a Configuration Manager site.

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

<CommonParameters>

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

Examples

Example 1: Get a site by using a site name

This command gets a System Center 2012 Configuration Manager site by using the site name CMSiteSystem.

```
PS C:\> Get-CMSite -SiteName "CMSiteSystem"
```

Related topics

[Set-CMSite](#)

Get-CMSiteInstallStatus

Get-CMSiteInstallStatus

Provides information about Configuration Manager installation status.

Syntax

Parameter Set: SearchBySiteCode

```
Get-CMSiteInstallStatus [-SiteCode <String> ] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Get-CMSiteInstallStatus -Id <String[]> [ <CommonParameters>]
```

Detailed Description

The **Get-CMSiteInstallStatus** cmdlet provides information about the installation status for Microsoft System Center 2012 Configuration Manager. You can specify an installation by ID or by site code.

Parameters

-Id<String[]>

Specifies an array of IDs for Configuration Manager installations.

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

-SiteCode<String>

Specifies a site code for a Configuration Manager site.

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

<CommonParameters>

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

Examples

Example 1: Get site installation status

This command gets the site installation status for the site that has the specified site code.

```
PS C:\> Get-CMSiteInstallStatus -SiteCode "CM1"
```

Get-CMSiteMaintenanceTask

Get-CMSiteMaintenanceTask

Gets maintenance tasks in Configuration Manager.

Syntax

Parameter Set: SearchByName

```
Get-CMSiteMaintenanceTask -SiteCode <String> [-MaintenanceTaskName <String[]> ] [ <CommonParameters> ]
```

Detailed Description

The **Get-CMSiteMaintenanceTask** cmdlet gets maintenance tasks in Microsoft System Center 2012 Configuration Manager. A maintenance task is a task in System Center 2012 Configuration Manager that performs maintenance on sites and databases.

Parameters

-MaintenanceTaskName<String[]>

Specifies an array of names for maintenance tasks.

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

-SiteCode<String>

Specifies the site code of the Configuration Manager site.

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

<CommonParameters>

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

Examples

Example 1: Get a maintenance task

This command gets the maintenance task named Backup for the Configuration Manager site that has the site code CM1.

```
PS C:\> Get-CMSiteMaintenanceTask -SiteCode "CM1" -MaintenanceTaskName "Backup"
```

Related topics

[Set-CMSiteMaintenanceTask](#)

Get-CMSiteStatusMessage

Get-CMSiteStatusMessage

Gets site system status messages.

Syntax

Parameter Set: SearchBySiteStatus

```
Get-CMSiteStatusMessage -ViewingPeriod <DateTime> [-ComputerName <String> ] [-Severity {All | Error | Information | Warning} ] [-SiteCode <String> ] [ <CommonParameters>]
```

Detailed Description

The **Get-CMSiteStatusMessage** cmdlet gets status messages for one or more Microsoft System Center 2012 Configuration Manager site system components. A status message is a message that a System Center 2012 Configuration Manager component generates that contains information about the success, failure, or operation of site system components. System Center 2012 Configuration Manager stores status messages in a System Center 2012 Configuration Manager site database. You can view status messages in the Status Message Viewer.

Parameters

-ComputerName<String>

Specifies the fully qualified domain name (FQDN) of the server that hosts the site system role.

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

-Severity<Severity>

Specifies the severity of a status message. Valid values are:

- All
- Error
- Information
- Warning

The acceptable values for this parameter are:

All	
Error	
Information	
Warning	

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

-SiteCode<String>

Specifies the site code of the Configuration Manager site that hosts the site system role.

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

-ViewingPeriod<DateTime>

Specifies the viewing period for site system status messages.

You can view only the status messages that Configuration Manager receives on or after the date and time that you specify. The default value is the current date, time, and time zone.

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

<CommonParameters>

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

Examples

Example 1: Get site status messages

This command gets the site status messages that System Center 2012 Configuration Manager receives on or after September 3, 2012 and that have an error severity. System Center 2012 Configuration Manager gets the status messages from the Configuration Manager site that has the site code CM2 on the computer named cmcen-dist02.tsqa.contoso.com.

```
PS C:\> Get-CMSiteStatusMessage -ViewingPeriod "2012/09/03 02:16:10.000" -ComputerName "cmcen-dist02" -Severity Error -SiteCode "CM2"
```

Get-CMSiteSystemServer

Get-CMSiteSystemServer

Retrieves an object that represents a site system server in System Center 2012 Configuration Manager.

Syntax

Parameter Set: SearchByName

```
Get-CMSiteSystemServer [-SiteCode <String> ] [-SiteSystemServerName <String> ] [ <CommonParameters> ]
```

Detailed Description

The **Get-CMSiteSystemServer** cmdlet retrieves an object that represents a site system server in Microsoft System Center 2012 Configuration Manager. A site system server provides functionality to a configuration management site, such as communication between a System Center 2012 Configuration Manager server and System Center 2012 Configuration Manager clients.

Parameters

-SiteCode<String>

Specifies a site code for the Configuration Manager site.

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

-SiteSystemServerName<String>

Specifies a server name for the Configuration Manager site.

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

Example 1: Get a site system server by using the site code and server name

The command uses the **Get-CMSiteSystemServer** cmdlet to get an object that represents the site system server.

```
PS C:\> Get-CMSiteSystemServer -SiteCode "CM2" -SiteSystemServerName "ContosoSvr02"
```

Related topics

[New-CMSiteSystemServer](#)

[Set-CMSiteSystemServer](#)

Get-CMSoftwareDistributionComponent

Get-CMSoftwareDistributionComponent

Gets an object that represents a software distribution component in Configuration Manager.

Syntax

Parameter Set: SearchBySiteCodeMandatory

```
Get-CMSoftwareDistributionComponent -SiteCode <String> [ <CommonParameters>]
```

Parameter Set: SearchByNameMandatory

```
Get-CMSoftwareDistributionComponent -SiteSystemServerName <String[]> [ <CommonParameters>]
```

Detailed Description

The **Get-CMSoftwareDistributionComponent** cmdlet gets an object that represents a software distribution component in Microsoft System Center 2012 Configuration Manager. A software distribution component consists of individual components, such as the client software distribution component.

Parameters

-SiteCode<String>

Specifies a site code of a Configuration Manager site.

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

-SiteSystemServerName<String[]>

Specifies an array of site system server names.

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

<CommonParameters>

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

Examples

Example 1: Get a software distribution component

This command gets the software distribution component.

```
PS C:\> Get-CMSoftwareDistributionComponent -Name "CM2"
```

Related topics

[Set-CMSoftwareDistributionComponent](#)

Get-CMSoftwareInventory

Get-CMSoftwareInventory

Retrieves an object that collects software inventory data in Configuration Manager.

Syntax

Parameter Set: SearchByName

```
Get-CMSoftwareInventory [-Name <String[]> ] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Get-CMSoftwareInventory -Id <String[]> [ <CommonParameters>]
```

Detailed Description

The **Get-CMSoftwareInventory** cmdlet retrieves an object that collects information about files that client devices contain.

Parameters

-Id<String[]>

Specifies an array of IDs of software files.

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

-Name<String[]>

Specifies an array of names of software files.

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

<CommonParameters>

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

Examples

Example 1: Get a software inventory object

This command gets the software inventory object named MSXML 6.0 Parser.

```
PS C:\> Get-CMSoftwareInventory -Name "MSXML 6.0 Parser"
```

Related topics

[Set-CMSoftwareInventory](#)

[Undo-CMSoftwareInventory](#)

Get-CMSoftwareMeteringRule

Get-CMSoftwareMeteringRule

Gets Configuration Manager software metering rules.

Syntax

Parameter Set: SearchByName

```
Get-CMSoftwareMeteringRule [-ProductName <String> ] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Get-CMSoftwareMeteringRule -Id <String[]> [ <CommonParameters>]
```

Detailed Description

The **Get-CMSoftwareMeteringRule** cmdlet gets one or more software metering rules in Microsoft System Center 2012 Configuration Manager. You can use this cmdlet to get rules to pass to other cmdlets, such as the **Enable-CMSoftwareMeteringRule** cmdlet or the **Remove-CMSoftwareMeteringRule** cmdlet.

Software metering monitors and collects software usage data from System Center 2012 Configuration Manager clients, such as when clients began using a particular software program and how long users have worked with that software. You can create software metering rules that specify which software to monitor.

You can specify rules by ID or by product name.

For more information about software metering in System Center 2012 Configuration Manager, see [Introduction to Software Metering in Configuration Manager](http://go.microsoft.com/fwlink/?LinkId=268432) (<http://go.microsoft.com/fwlink/?LinkId=268432>) on TechNet.

Parameters

-Id<String[]>

Specifies an array of IDs for software metering rules.

Aliases	RuleId
Required?	true
Position?	named

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

-ProductName<String>

Specifies a name for a product that a rule meters.

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

<CommonParameters>

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

Examples

Example 1: Get rules for a product

This command gets software metering rules for the product named Accounting Package.

```
PS C:\> Get-CMSoftwareMeteringRule -ProductName "Accounting Package"
```

Related topics

[Disable-CMSoftwareMeteringRule](#)

[Enable-CMSoftwareMeteringRule](#)

[New-CMSoftwareMeteringRule](#)

[Remove-CMSoftwareMeteringRule](#)

[Set-CMSoftwareMeteringRule](#)

Get-CMSoftwareMeteringSetting

Get-CMSoftwareMeteringSetting

Gets Configuration Manager software metering settings.

Syntax

```
Get-CMSoftwareMeteringSetting [ <CommonParameters>]
```

Detailed Description

The **Get-CMSoftwareMeteringSetting** cmdlet gets a software metering settings object in Microsoft System Center 2012 Configuration Manager.

Parameters

<CommonParameters>

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

Examples

Example 1: Get software metering setting object

This command gets a software metering setting object.

```
PS C:\> Get-CMSoftwareMeteringSetting
```

```
ClientComponentName : Software Metering Agent
FileType             : 2
Flags                : 1
ItemName             : Software Metering Agent
ItemType             : Client Component
PropLists            :
Props                :
RegMultiStringLists :
```

SiteCode : CM1

Related topics

[Set-CMSoftwareMeteringSetting](#)

Get-CMSoftwareUpdate

Get-CMSoftwareUpdate

Retrieves configuration settings for software updates.

Syntax

Parameter Set: SearchByName

```
Get-CMSoftwareUpdate [-Name <String[]> ] [-SecuredScopeNames <String> ] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Get-CMSoftwareUpdate -Id <String[]> [-SecuredScopeNames <String> ] [ <CommonParameters>]
```

Parameter Set: SearchByUpdateGroup

```
Get-CMSoftwareUpdate -UpdateGroup <IResultObject> [ <CommonParameters>]
```

Parameter Set: SearchByUpdateGroupIdMandatory

```
Get-CMSoftwareUpdate -UpdateGroupId <String[]> [ <CommonParameters>]
```

Parameter Set: SearchByUpdateGroupNameMandatory

```
Get-CMSoftwareUpdate -UpdateGroupName <String[]> [ <CommonParameters>]
```

Detailed Description

The **Get-CMSoftwareUpdate** cmdlet retrieves configuration settings for one or more software updates. Clients receive a software update object when manually or automatically deploying a software update.

Parameters

-Id<String[]>

Specifies an array of IDs of software updates.

Aliases	CId
Required?	true
Position?	named
Default Value	none

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

-Name<String[]>

Specifies an array of names of software updates.

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

-SecuredScopeNames<String>

Specifies a security scope for the deployment package. The default value is Default.

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

-UpdateGroup<IResultObject>

Specifies a **CMSoftwareUpdate** object.

Aliases	none
Required?	true
Position?	named
Default Value	none

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

-UpdateGroupId<String[]>

Specifies an array of IDs of update groups.

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

-UpdateGroupName<String[]>

Specifies an array of names of update groups.

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

<CommonParameters>

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

Examples

Example 1: Retrieve all software updates

This command retrieves all software updates.

```
PS C:\> Get-CMSoftwareUpdate
```

Example 2: Retrieve a software update by using a name

This command retrieves the software update named Cumulative Security Update for ActiveX Killbits for Windows 7 (KB2618451).

```
PS C:\> Get-CMSoftwareUpdate -Name "Cumulative Security Update for ActiveX Killbits for Windows 7 (KB2618451)"
```

Related topics

[Save-CMSoftwareUpdate](#)

[Set-CMSoftwareUpdate](#)

[Sync-CMSoftwareUpdate](#)

Get-CMSoftwareUpdateAutoDeploymentRule

Get-CMSoftwareUpdateAutoDeploymentRule

Gets Configuration Manager deployment rules for automatic software updates.

Syntax

Parameter Set: SearchByName

```
Get-CMSoftwareUpdateAutoDeploymentRule [-Name <String> ] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Get-CMSoftwareUpdateAutoDeploymentRule -Id <String[]> [ <CommonParameters>]
```

Detailed Description

The **Get-CMSoftwareUpdateAutoDeploymentRule** cmdlet gets specified Microsoft System Center 2012 Configuration Manager deployment rules for automatic software updates.

System Center 2012 Configuration Manager uses rules to manage automatic deployment of software updates. When a rule runs, System Center 2012 Configuration Manager adds updates that qualify for the rule to a software update group. The System Center 2012 Configuration Manager server downloads content files and copies them to distribution points, and then updates client computers.

You can specify rules by ID or by name. You can use this cmdlet to get deployment rules for automatic software updates to use with other cmdlets, such as the **Invoke-CMSoftwareUpdateAutoDeploymentRule** cmdlet or the **Remove-CMSoftwareUpdateAutoDeploymentRule** cmdlet.

Parameters

-Id<String[]>

Specifies an array of IDs for rules for automatic deployment of software updates. This value is the **AutoDeploymentID** property of the deployment rule object.

Aliases	AutoDeploymentId
Required?	true
Position?	named

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

-Name<String>

Specifies a name of a rule for automatic deployment of software updates.

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

<CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about_CommonParameters](#).

Examples

Example 1: Get a deployment rule by name

This command gets a deployment rule named Weekly Driver Updates.

```
PS C:\> Get-CMSoftwareUpdateAutoDeploymentRule -Name "Weekly Driver Updates"
```

Example 2: Get a deployment rule by ID

This command gets a deployment rule that has the ID 16777217.

```
PS C:\> Get-CMSoftwareUpdateAutoDeploymentRule -Id "16777217"
```

Related topics

[Disable-CMSoftwareUpdateAutoDeploymentRule](#)

[Enable-CMSoftwareUpdateAutoDeploymentRule](#)

[Invoke-CMSoftwareUpdateAutoDeploymentRule](#)

[New-CMSoftwareUpdateAutoDeploymentRule](#)

[Remove-CMSoftwareUpdateAutoDeploymentRule](#)

[Set-CMSoftwareUpdateAutoDeploymentRule](#)

Get-CMSoftwareUpdateBasedClientInstallation

Get-CMSoftwareUpdateBasedClientInstallation

Gets a client installation on a Configuration Manager software update point.

Syntax

Parameter Set: SearchByNameMandatory

```
Get-CMSoftwareUpdateBasedClientInstallation -SiteSystemServerName <String[]> [  
<CommonParameters>]
```

Parameter Set: SearchBySiteCodeMandatory

```
Get-CMSoftwareUpdateBasedClientInstallation -SiteCode <String> [-SiteSystemServerName  
<String[]> ] [ <CommonParameters>]
```

Detailed Description

The **Get-CMSoftwareUpdateBasedClientInstallation** cmdlet gets a client installation hosted on a software update point for Microsoft System Center 2012 Configuration Manager.

System Center 2012 Configuration Manager publishes the System Center 2012 Configuration Manager client to a software update point. This site system role can install the client on computers that do not already have it or upgrade existing clients.

To use software update point based installation, you must use the same Windows Server Update Services (WSUS) server for both client installation and software updates. This server must be the active software update point in a primary site.

Parameters

-SiteCode<String>

Specifies a site code for a Configuration Manager site.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-SiteSystemServerName<String[]>

Specifies an array of names of servers that host a software update point role.

Aliases	Name
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

<CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about_CommonParameters](#).

Examples

Example 1: Get a client installation

This command gets the client installation for the site that has the site code CM1.

```
PS C:\> Get-CMSoftwareUpdateBasedClientInstallation -SiteCode "CM1"
```

Related topics

[Set-CMSoftwareUpdateBasedClientInstallation](#)

Get-CMSoftwareUpdateDeploymentPackage

Get-CMSoftwareUpdateDeploymentPackage

Retrieves a deployment package.

Syntax

Parameter Set: SearchByName

```
Get-CMSoftwareUpdateDeploymentPackage [-Name <String> ] [-SecuredScopeNames <String> ] [ <CommonParameters> ]
```

Parameter Set: SearchByIdMandatory

```
Get-CMSoftwareUpdateDeploymentPackage -Id <String[]> [-SecuredScopeNames <String> ] [ <CommonParameters> ]
```

Detailed Description

The **Get-CMSoftwareUpdateDeploymentPackage** cmdlet retrieves a deployment package for a software update. A **CMSoftwareUpdateDeploymentPackage** object contains one or more software updates.

Parameters

-Id<String[]>

Specifies an array of identifiers for the deployment package.

Aliases	PackageId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies a name for the deployment package.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SecuredScopeNames<String>

Specifies an array of security scopes for the deployment package. The default value is Default.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

<CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

Examples

Example 1: Get a deployment package by using a name

This command gets a deployment package named Asdset.

```
PS C:\> Get-CMSoftwareUpdateDeploymentPackage -Name "Asdset"
```

Example 2: Get a deployment package by using an ID

This command gets a deployment package that has the ID ST10000C.

```
PS C:\> Get-CMSoftwareUpdateDeploymentPackage -Id "ST10000C"
```

Related topics

[Remove-CMSoftwareUpdateDeploymentPackage](#)

[Set-CMSoftwareUpdateDeploymentPackage](#)

Get-CMSoftwareUpdateGroup

Get-CMSoftwareUpdateGroup

Gets software update groups.

Syntax

Parameter Set: SearchByName

```
Get-CMSoftwareUpdateGroup [-Name <String[]> ] [-SecuredScopeNames <String> ] [ <CommonParameters> ]
```

Parameter Set: SearchByIdMandatory

```
Get-CMSoftwareUpdateGroup -Id <String[]> [-SecuredScopeNames <String> ] [ <CommonParameters> ]
```

Detailed Description

The **Get-CMSoftwareUpdateGroup** cmdlet gets one or more software update groups in Microsoft System Center 2012 Configuration Manager. A software update group is a collection of one or more software updates. You can add software updates to a software update group and then deploy the group to clients. After you deploy a software update group, you can add new software updates to the group and System Center 2012 Configuration Manager automatically deploys them.

Parameters

-Id<String[]>

Specifies an array of IDs of software update groups.

Aliases	CIId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String[]>

Specifies an array of names of software update groups.

Aliases	LocalizedDisplayName
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SecuredScopeNames<String>

Specifies a security scope name. A security scope name can be either Default or the name of a custom security scope.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

<CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about_CommonParameters](#).

Examples

Example 1: Get software update groups

This command gets all software update groups.

```
PS C:\> Get-CMSoftwareUpdateGroup
```

Example 2: Get a software update group by using an ID

This command gets a software update group that has the ID ST10000D.

```
PS C:\> Get-CMSoftwareUpdateGroup -Id "ST10000D"
```

Example 3: Get a software update group by using a name

This command gets a software update group object named SUGroupD01.

```
PS C:\> Get-CMSoftwareUpdateGroup CMSoftwareUpdateGroup -Name "SUGroupD01"
```

Related topics

[New-CMSoftwareUpdateGroup](#)

[Remove-CMSoftwareUpdateGroup](#)

[Set-CMSoftwareUpdateGroup](#)

Get-CMSoftwareUpdateLicense

Get-CMSoftwareUpdateLicense

Gets a EULA or SLT for a software update in Configuration Manager.

Syntax

Parameter Set: SearchByName

```
Get-CMSoftwareUpdateLicense [-Name <String[]> ] [-SecuredScopeNames <String> ] [ <CommonParameters> ]
```

Parameter Set: SearchByIdMandatory

```
Get-CMSoftwareUpdateLicense -Id <String[]> [-SecuredScopeNames <String> ] [ <CommonParameters> ]
```

Parameter Set: SearchByValueMandatory

```
Get-CMSoftwareUpdateLicense -InputObject <IResultObject> [ <CommonParameters> ]
```

Detailed Description

The **Get-CMSoftwareUpdateLicense** cmdlet gets an End User License Agreement (EULA) or Software License Terms (SLT) for a software update in Microsoft System Center 2012 Configuration Manager. You can specify a software update by ID or by name or use the **Get-CMSoftwareUpdate** cmdlet to obtain one. If you specify an ID or name, you can further specify a security scope membership.

Parameters

-Id<String[]>

Specifies an array of IDs for software updates.

Aliases	CId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-InputObject<IResultObject>

Specifies a software update object. To obtain a software update object, use the **Get-CMSoftwareUpdate** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String[]>

Specifies an array of names of software updates.

Aliases	LocalizedDisplayName
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SecuredScopeNames<String>

Specifies a security scope membership.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

<CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about_CommonParameters](#).

Examples

Example 1: Get a EULA or SLT for a software update

This command gets the EULA or SLT for a software update named UpdatePkg01 for the security scope named SecScope02.

```
PS C:\> Get-CMSoftwareUpdateLicense -Name "UpdatePkg01" -SecuredScopeNames "SecScope02"
```

Related topics

[Get-CMSoftwareUpdate](#)

Get-CMSoftwareUpdatePoint

Get-CMSoftwareUpdatePoint

Gets a Configuration Manager software update point.

Syntax

Parameter Set: SearchByName

```
Get-CMSoftwareUpdatePoint [-SiteCode <String> ] [-SiteSystemServerName <String> ] [ <CommonParameters> ]
```

Detailed Description

The **Get-CMSoftwareUpdatePoint** cmdlet gets a software update point site system role for Microsoft System Center 2012 Configuration Manager.

A software update point is a site server role that hosts software updates. System Center 2012 Configuration Manager clients connect to a software update point to get available updates. The software update point interacts with Windows Server Update Services (WSUS) to configure update settings, request synchronization to the update source, and to synchronize software updates from the WSUS database.

You can specify a software update point by site code or by the name of the computer that hosts the site system role.

Parameters

-SiteCode<String>

Specifies a site code for a Configuration Manager site.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteSystemServerName<String>

Specifies the name of a computer that hosts the software update point site system role.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

<CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about_CommonParameters](#).

Examples

Example 1: Get a software update point

The command gets a software update point that UpdateSystem.Western.Contoso.com hosts.

```
PS C:\> Get-CMSoftwareUpdatePoint -SiteSystemServerName "UpdateSystem.Western.Contoso.com"
```

Related topics

[Add-CMSoftwareUpdatePoint](#)

[Remove-CMSoftwareUpdatePoint](#)

[Set-CMSoftwareUpdatePoint](#)

Get-CMSoftwareUpdatePointComponent

Get-CMSoftwareUpdatePointComponent

Retrieves a software update point component in Configuration Manager.

Syntax

Parameter Set: SearchByNameMandatory

```
Get-CMSoftwareUpdatePointComponent -SiteSystemServerName <String[]> [ <CommonParameters>]
```

Parameter Set: SearchBySiteCodeMandatory

```
Get-CMSoftwareUpdatePointComponent -SiteCode <String> [-SiteSystemServerName <String[]> ] [ <CommonParameters>]
```

Detailed Description

The **Get-CMSoftwareUpdatePointComponent** cmdlet retrieves a software update point component. A software update point component interacts with WSUS services to configure update settings, request synchronization to the upstream update source, and synchronize updates from the WSUS database to the site server database on the central site.

Parameters

-SiteCode<String>

Specifies a site code in Configuration Manager.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteSystemServerName<String[]>

Specifies an array of names of a site system servers in Configuration Manager.

Aliases	Name
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

<CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about_CommonParameters](#).

Examples

Example 1: Retrieve a software update point component by name

This command retrieves a software update point component by using the site system server name.

```
PS C:\> Get-CMSoftwareUpdatePointComponent -SiteSystemServerName "Contoso-  
SiteSysSrv.Western.Contoso.com"
```

Example 2: Retrieve a software update point component by site code

This command retrieves a software update point component by using the site code.

```
PS C:\> Get-CMSoftwareUpdatePointComponent -SiteCode "CM1"
```

Related topics

[Set-CMSoftwareUpdatePointComponent](#)

Get-CMSoftwareUpdateSummarizationSchedule

Get-CMSoftwareUpdateSummarizationSchedule

Displays the Configuration Manager schedule for software update summarization.

Syntax

```
Get-CMSoftwareUpdateSummarizationSchedule [ <CommonParameters> ]
```

Detailed Description

The **Get-CMSoftwareUpdateSummarizationSchedule** cmdlet displays the current schedule for software update summarization for Microsoft System Center 2012 Configuration Manager. You can use the **Set-CMSoftwareUpdateSummarizationSchedule** cmdlet to change the schedule. You can use the **Invoke-CMSoftwareUpdateSummarization** cmdlet to run the summarization immediately.

Parameters

<CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about_CommonParameters](#).

Examples

Example 1: Display the summarization schedule

This command displays the summarization schedule for software updates. In this case, the schedule is every 12 hours.

```
PS C:\> Get-CMSoftwareUpdateSummarizationSchedule
```

Interval	Unit
-----	----
12	Hours

Related topics

[Set-CMSoftwareUpdateSummarizationSchedule](#)

[Invoke-CMSoftwareUpdateSummarization](#)

Get-CMStateMigrationPoint

Get-CMStateMigrationPoint

Gets a state migration point for a Configuration Manager site.

Syntax

Parameter Set: SearchByName

```
Get-CMStateMigrationPoint [-SiteCode <String> ] [-SiteSystemServerName <String> ] [ <CommonParameters> ]
```

Detailed Description

The **Get-CMStateMigrationPoint** cmdlet gets a state migration point for a Microsoft System Center 2012 Configuration Manager site. This site system role stores user information. You can store user information while an operating system deployment proceeds and then restore user information from the state migration point.

You can use this cmdlet to get state migration point objects to use with the **Remove-CMStateMigrationPoint** cmdlet.

Each state migration point site server can be a member of only one System Center 2012 Configuration Manager site.

Parameters

-SiteCode<String>

Specifies a site code for a Configuration Manager site.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteSystemServerName<String>

Specifies the host name for a state migration point.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

<CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about_CommonParameters](#).

Examples

Example 1: Get a migration point

This command gets a state migration point that belongs to the specified site and has the specified host name.

```
PS C:\> Get-CMStateMigrationPoint -SiteCode "CM1" -SiteSystemServerName  
"SMP01.Western.Contoso.com"
```

Related topics

[Add-CMStateMigrationPoint](#)

[Remove-CMStateMigrationPoint](#)

[Set-CMStateMigrationPoint](#)

Get-CMStatusFilterRule

Get-CMStatusFilterRule

Gets Configuration Manager filter rules for status messages.

Syntax

Parameter Set: SearchBySiteCodeMandatory

```
Get-CMStatusFilterRule -SiteCode <String> [-Name <String> ] [ <CommonParameters>]
```

Detailed Description

The **Get-CMStatusFilterRule** cmdlet gets filter rules for Microsoft System Center 2012 Configuration Manager status messages. You can get all the rules for a System Center 2012 Configuration Manager site or you can specify a name of a rule within a site.

Status filter rules specify how System Center 2012 Configuration Manager responds to status messages. Each filter rule contains criteria and actions for status messages. You configure status filter rules for each site, not across all sites.

Parameters

-Name<String>

Specifies a name of a rule.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteCode<String>

Specifies a site code for the Configuration Manager site.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

<CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about_CommonParameters](#).

Examples

Example 1: Get rules for a specified site

This cmdlet gets status filter rules for the site that has the site code CM1.

```
PS C:\> Get-CMStatusFilterRule -SiteCode "CM1"
```

Related topics

[Disable-CMStatusFilterRule](#)

[Enable-CMStatusFilterRule](#)

[New-CMStatusFilterRule](#)

[Remove-CMStatusFilterRule](#)

[Set-CMStatusFilterRule](#)

Get-CMStatusMessageQuery

Get-CMStatusMessageQuery

Gets Configuration Manager status message queries or displays messages.

Syntax

Parameter Set: SearchByName

```
Get-CMStatusMessageQuery [-Name <String> ] [-ShowMessages] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Get-CMStatusMessageQuery -Id <String> [-ShowMessages] [ <CommonParameters>]
```

Detailed Description

The **Get-CMStatusMessageQuery** cmdlet gets Microsoft System Center 2012 Configuration Manager status message queries. Status message queries return status messages from a System Center 2012 Configuration Manager site database. You can use this cmdlet with the *ShowMessages* parameter to display messages found by this query.

You can use this cmdlet to get queries to use with the **Set-CMStatusMessageQuery** cmdlet or the **Remove-CMStatusMessageQuery** cmdlet.

Parameters

-Id<String>

Specifies an ID of a status message query.

Aliases	QueryId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies a name of a status message query.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ShowMessages

Indicates that the cmdlet shows messages. This cmdlet opens the Status Message Viewer.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

<CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

Examples

Example 1: Get a query that has a specified name

This command gets a query that has a specified name.

```
PS C:\> Get-CMStatusMessageQuery -Name "Clients That Received a Specific Deployed Program"
```

Example 2: Show messages for a query

This command shows messages found by a query that has an ID of SMS551.

```
PS C:\> Get-CMStatusMessageQuery -Id "SMS551" -ShowMessages
```

Related topics

[New-CMStatusMessageQuery](#)

[Remove-CMStatusMessageQuery](#)

[Set-CMStatusMessageQuery](#)

Get-CMStatusReportingComponent

Get-CMStatusReportingComponent

Gets an object representing a status reporting component.

Syntax

Parameter Set: SearchBySiteCodeMandatory

```
Get-CMStatusReportingComponent -SiteCode <String> [ <CommonParameters>]
```

Parameter Set: SearchByName

```
Get-CMStatusReportingComponent -SiteSystemServerName <String[]> [ <CommonParameters>]
```

Detailed Description

The **Get-CMStatusReportingComponent** cmdlet gets an object that represents the status reporting component. This object provides information about the client configuration and server configuration components.

Parameters

-SiteCode<String>

Specifies a site code of a Configuration Manager site.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteSystemServerName<String[]>

Specifies an array of site system server names.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

<CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about_CommonParameters](#).

Examples

Example 1: Get status reporting components

This command gets the status reporting components that **Set-CMStatusReportingComponent** configures for the site.

```
PS C:\> Get-CMStatusReportingComponent -SiteCode "CM1"
```

Related topics

[Set-CMStatusReportingComponent](#)

Get-CMStatusSummarizer

Get-CMStatusSummarizer

Gets a status summarizer object for Configuration Manager.

Syntax

Parameter Set: SearchBySiteCodeMandatory

```
Get-CMStatusSummarizer -SiteCode <String> -StatusSummarizerType  
{ApplicationDeploymentSummarizer | ApplicationStatisticsSummarizer |  
ComponentStatusSummarizer | SiteSystemStatusSummarizer} [ <CommonParameters>]
```

Detailed Description

The **Get-CMStatusSummarizer** cmdlet gets a status summarizer object. The Microsoft System Center 2012 Configuration Manager status summarizers apply to the areas of application deployment, application statistics, component status, and site system status.

Parameters

-SiteCode<String>

Specifies a site code for the Configuration Manager site.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-StatusSummarizerType<StatusSummarizerType>

Specifies a status summarization type.

The acceptable values for this parameter are:

ApplicationDeploymentSummarizer	
ApplicationStatisticsSummarizer	
ComponentStatusSummarizer	
SiteSystemStatusSummarizer	

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

<CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about_CommonParameters](#).

Examples

Example 1: Get a status summarizer

This command gets the status summarizer for the component status.

```
PS C:\> Get-CMStatusSummarizer -SiteCode "CM1" -StatusSummarizerType
ComponentStatusSummarizer
```

Related topics

[Set-CMStatusSummarizer](#)

Get-CMSystemHealthValidatorPoint

Get-CMSystemHealthValidatorPoint

Gets a system health validator point for Configuration Manager.

Syntax

Parameter Set: SearchByName

```
Get-CMSystemHealthValidatorPoint [-SiteCode <String> ] [-SiteSystemServerName <String> ] [ <CommonParameters> ]
```

Detailed Description

The **Get-CMSystemHealthValidatorPoint** cmdlet gets a system health validator point for a Microsoft System Center 2012 Configuration Manager site. This site system role validates statements of health from a server that is running Network Policy Server (NPS).

You can specify a validator point by site system name, site code, or both. You can use this cmdlet with the **Remove-CMSystemHealthValidatorPoint** cmdlet.

Parameters

-SiteCode<String>

Specifies a site code for a Configuration Manager site.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteSystemServerName<String>

Specifies the host name for a system health validator point.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

<CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about_CommonParameters](#).

Examples

Example 1: Get a validator point

This command gets a system health validator point. The command specifies the site code and the name of the server that hosts that system role.

```
PS C:\> Get-CMSystemHealthValidatorPoint -SiteCode "CM1" -SiteSystemServerName  
"Test01.Western.Contoso.com"
```

Related topics

[Add-CMSystemHealthValidatorPoint](#)

[Remove-CMSystemHealthValidatorPoint](#)

Get-CMSystemHealthValidatorPointComponent

Get-CMSystemHealthValidatorPointComponent

Retrieves an object that represents a system health validator point in Configuration Manager.

Syntax

Parameter Set: SearchByNameMandatory

```
Get-CMSystemHealthValidatorPointComponent -SiteSystemServerName <String[]> [  
<CommonParameters>]
```

Parameter Set: SearchBySiteCodeMandatory

```
Get-CMSystemHealthValidatorPointComponent -SiteCode <String> [ <CommonParameters>]
```

Detailed Description

The **Get-CMSystemHealthValidatorPointComponent** cmdlet retrieves an object that represents a system health validator point. A system health validator point is a site system role that evaluates system health information reported by Windows clients for security related compliance.

Parameters

-SiteCode<String>

Specifies a site code in Configuration Manager.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteSystemServerName<String[]>

Specifies an array of site system server names in Configuration Manager.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

<CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about_CommonParameters](#).

Examples

Example 1: Retrieve a system health validator point by site system server name

This command retrieves a system health validator point component by using a site system server name.

```
PS C:\> Get-CMSystemHealthValidatorPointComponent -SiteSystemServerName "Shvp-01.Western.Corp.Contoso.com"
```

Example 2: Retrieve a system health validator point by site code

This command retrieves a system health validator point component by using a site code.

```
PS C:\> Get-CMSystemHealthValidatorPointComponent -SiteCode "CM4"
```

Related topics

[Set-CMSystemHealthValidatorPointComponent](#)

Get-CMTaskSequence

Get-CMTaskSequence

Gets Configuration Manager task sequences.

Syntax

Parameter Set: SearchByName

```
Get-CMTaskSequence [-Name <String> ] [-SecuredScopeNames <String> ] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Get-CMTaskSequence -TaskSequencePackageId <String[]> [-SecuredScopeNames <String> ] [ <CommonParameters>]
```

Detailed Description

The **Get-CMTaskSequence** cmdlet gets Microsoft System Center 2012 Configuration Manager task sequences. A task sequence includes configuration and operating system deployment settings for a System Center 2012 Configuration Manager client computer.

You can specify a name or ID to get a specific sequence. You can also specify a security scope, by itself or with a name or ID, to get sequences with that security scope.

Parameters

-Name<String>

Specifies a name for a task sequence.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SecuredScopeNames<String>

Specifies a name of a security scope.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-TaskSequencePackageld<String[]>

Specifies an array of IDs of task sequences.

Aliases	Packageld
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

<CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

Examples

Example 1: Get task sequences

This command gets all task sequence objects for the current System Center 2012 Configuration Manager.

```
PS C:\> Get-CMTaskSequence
```

Example 2: Get a task sequence by using a ID

This command gets the task sequence that has an ID of CM100015.

```
PS C:\> Get-CMTaskSequence -Id "CM100015"
```

Example 3: Get task sequence by using a name

This command gets the task sequence named General Sequence if it belongs to the security scope called Scope22. A task sequence can belong to more than one security scope.

```
PS C:\> Get-CMTaskSequence -Name "General Sequence" -SecuredScopeNames "Scope22"
```

Related topics

[Disable-CMTaskSequence](#)

[Enable-CMTaskSequence](#)

[Export-CMTaskSequence](#)

[Import-CMTaskSequence](#)

[New-CMTaskSequence](#)

[Remove-CMTaskSequence](#)

[Set-CMTaskSequence](#)

Get-CMTrustedRootCertificate

Get-CMTrustedRootCertificate

Gets a trusted root certificate for Configuration Manager.

Syntax

Parameter Set: GetByName

```
Get-CMTrustedRootCertificate -CertificationAuthorityServerName <String> [  
<CommonParameters>]
```

Detailed Description

The **Get-CMTrustedRootCertificate** cmdlet gets a trusted root certificate for Microsoft System Center 2012 Configuration Manager. For native mode communication, System Center 2012 Configuration Manager authenticates, encrypts, and signs communications based on public key infrastructure (PKI) keys that depend on trusted root certificate. Devices that communicate by using certificates must have a root certificate in common. Devices in your System Center 2012 Configuration Manager hierarchy might have different root certificates. If so, install all necessary trusted root certificates.

Computers that run the Windows operating system, as well as many other devices, rely on some well-known third-party root certificates. If you deploy your own PKI, install the required root certificate.

Parameters

-CertificationAuthorityServerName<String>

Specifies the name of a Certification Authority server.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

<CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about_CommonParameters](#).

Examples

Example 1: Get a trusted root certificate

This command gets a trusted root certificate from the internal server named ContosoCA.Contoso.com.

```
PS C:\> Get-CTrustedReaderCertificate -CertificationAuthorityServerName  
"ContosoCA.Contoso.com"
```

Get- CMTrustedRootCertificateProfileConfigurationItem

Get-CMTrustedRootCertificateProfileConfigurationItem

Gets root certificate profiles.

Syntax

Parameter Set: SearchByName

```
Get-CMTrustedRootCertificateProfileConfigurationItem [-Name <String[]> ] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Get-CMTrustedRootCertificateProfileConfigurationItem -Id <String[]> [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Get-CMTrustedRootCertificateProfileConfigurationItem -InputObject <IResultObject> [ <CommonParameters>]
```

Detailed Description

The **Get-CMTrustedRootCertificateProfileConfigurationItem** cmdlet gets root certificate profiles.

Parameters

-Id<String[]>

Specifies an array of IDs of root certificate profiles.

Aliases	CId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-InputObject<IResultObject>

Specifies a root certificate profile object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String[]>

Specifies an array of names of root certificate profiles.

Aliases	LocalizedDisplayName
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

<CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about_CommonParameters](#).

Examples

Example 1: Get certificate profiles

This command gets all trusted root certificate profiles.

```
PS C:\> Get-CMTrustedRootCertificateProfileConfigurationItem
```

Related topics

[Copy-CMTrustedRootCertificateProfileConfigurationItem](#)

[New-CMTrustedRootCertificateProfileConfigurationItem](#)

[Remove-CMTrustedRootCertificateProfileConfigurationItem](#)

[Set-CMTrustedRootCertificateProfileConfigurationItem](#)

Get-CMUser

Get-CMUser

Gets a user of the Configuration Manager hierarchy.

Syntax

Parameter Set: ByName

```
Get-CMUser [-Name <String> ] [ <CommonParameters>]
```

Parameter Set: ById

```
Get-CMUser -Id <String> [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Get-CMUser -CollectionId <String> [-Name <String> ] [ <CommonParameters>]
```

Parameter Set: SearchByNameMandatory

```
Get-CMUser -CollectionName <String> [-Name <String> ] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Get-CMUser -Collection <IResultObject> [-Name <String> ] [ <CommonParameters>]
```

Detailed Description

The **Get-CMUser** cmdlet retrieves a user object that can be added to a collection.

To add a user to a collection use the [Add-CMUserCollectionDirectMembershipRule](#) cmdlet. For more information about Microsoft System Center 2012 Configuration Manager collections, see [Introduction to Collections in Configuration Manager](#) (<http://go.microsoft.com/fwlink/p/?LinkID=259433>) on TechNet.

Parameters

-Collection<IResultObject>

Specifies a Configuration Manager user collection object. To obtain a user collection object, use the **Get-CMUserCollection** cmdlet.

Aliases	none
Required?	true

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CollectionId<String>

Specifies the ID of a user collection in Configuration Manager.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CollectionName<String>

Specifies the name of a user collection in Configuration Manager.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Id<String>

Specifies the ID of the user to retrieve.

Aliases	ResourceID
Required?	true

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies the name of the user to retrieve.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

<CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about_CommonParameters](#).

Examples

Example 1: Get a user

This command gets the user of the Configuration Manager hierarchy named DChew.

```
PS C:\> Get-CMUser -Name "CENTRAL\DChew"
```

Related topics

[Remove-CMUser](#)

[Get-CMUserCollection](#)

Get-CMUserCollection

Get-CMUserCollection

Gets one or more user collections in the Configuration Manager hierarchy.

Syntax

Parameter Set: SearchByDPGroupId

```
Get-CMUserCollection -DistributionPointGroupId <String> [ <CommonParameters>]
```

Parameter Set: SearchByDPGroupName

```
Get-CMUserCollection -DistributionPointGroupName <String> [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Get-CMUserCollection -CollectionId <String> [ <CommonParameters>]
```

Parameter Set: SearchByName

```
Get-CMUserCollection [-Name <String> ] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Get-CMUserCollection -DistributionPointGroup <IResultObject> [ <CommonParameters>]
```

Detailed Description

The **Get-CMUserCollection** cmdlet retrieves collections that contain users in Microsoft System Center 2012 Configuration Manager. For more information about collections, see [Introduction to Collections in Configuration Manager](http://go.microsoft.com/fwlink/p/?LinkID=259433) (http://go.microsoft.com/fwlink/p/?LinkID=259433) on TechNet.

Parameters

-CollectionId<String>

Specifies the IDs of the user collections.

Aliases	none
Required?	true
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DistributionPointGroup<IResultObject>

Specifies an object that represents a distribution point group that is associated with the user collections.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DistributionPointGroupId<String>

Specifies the ID of a distribution point group that is associated with the user collections.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DistributionPointGroupName<String>

Specifies the name of a distribution point group that is associated with the user collections.

Aliases	none
Required?	true
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies the names of the user collections. If a specific collection is not specified, all user collections in the hierarchy are returned.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

<CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about_CommonParameters](#).

Examples

Example 1: Get a user collection

This command gets the user collection that has the ID 9990000D.

```
PS C:\> Get-CMUserCollection -CollectionId "9990000D"
```

Related topics

[Export-CMUserCollection](#)

[Import-CMUserCollection](#)

[New-CMUserCollection](#)

[Remove-CMUserCollection](#)

[Set-CMUserCollection](#)

Get-CMUserCollectionDirectMembershipRule

Get-CMUserCollectionDirectMembershipRule

Gets the direct membership rules of one or more user collections in the Configuration Manager hierarchy.

Syntax

Parameter Set: ByCollectionNameAndResourceName

```
Get-CMUserCollectionDirectMembershipRule -CollectionName <String> -ResourceName <String> [  
<CommonParameters>]
```

Parameter Set: ByCollectionIdAndResourceId

```
Get-CMUserCollectionDirectMembershipRule -CollectionId <String> -ResourceId <Int32> [  
<CommonParameters>]
```

Parameter Set: ByCollectionIdAndResourceName

```
Get-CMUserCollectionDirectMembershipRule -CollectionId <String> -ResourceName <String> [  
<CommonParameters>]
```

Parameter Set: ByCollectionNameAndResourceId

```
Get-CMUserCollectionDirectMembershipRule -CollectionName <String> -ResourceId <Int32> [  
<CommonParameters>]
```

Parameter Set: ByCollectionValueAndResourceId

```
Get-CMUserCollectionDirectMembershipRule -Collection <IResultObject> -ResourceId <Int32> [  
<CommonParameters>]
```

Parameter Set: ByCollectionValueAndResourceName

```
Get-CMUserCollectionDirectMembershipRule -Collection <IResultObject> -ResourceName <String>  
[ <CommonParameters>]
```

Detailed Description

The **Get-CMUserCollectionDirectMembershipRule** cmdlet retrieves the direct rules of the specified collections. You can specify the user collections by using their names, IDs, or by specifying an object that represents the collections.

A direct rule lets you explicitly choose the members of the user collection. For more information about collection rules, see [Introduction to Collections in Configuration Manager](http://go.microsoft.com/fwlink/p/?LinkID=259433) (<http://go.microsoft.com/fwlink/p/?LinkID=259433>) on TechNet.

Parameters

-Collection<IResultObject>

Specifies a Configuration Manager user collection object. To obtain a user collection object, use the **Get-CMUserCollection** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CollectionId<String>

Specifies the ID of a user collection.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CollectionName<String>

Specifies the name of a user collection.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-ResourceId<Int32>

Specifies the ID of the rule that you want to retrieve.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ResourceName<String>

Specifies the name of the rule that you want to retrieve.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

<CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about_CommonParameters](#).

Examples

Example 1: Get a direct membership rule

This command gets the direct membership rule that has the Id Res_94412512 for the collection named All Mobile Devices.

```
PS C:\> Get-CMUserCollectionDirectMembershipRule -CollectionName "All Mobile Devices" -  
ResourceId "Res_94412512"
```

Related topics

[Get-CMUserCollection](#)

[Add-CMUserCollectionDirectMembershipRule](#)

[Remove-CMUserCollectionDirectMembershipRule](#)

Get-CMUserCollectionExcludeMembershipRule

Get-CMUserCollectionExcludeMembershipRule

Gets the exclude membership rules from one or more user collections in the Configuration Manager hierarchy.

Syntax

Parameter Set: ByCollectionNameAndExcludeCollectionName

```
Get-CMUserCollectionExcludeMembershipRule -CollectionName <String> -ExcludeCollectionName <String> [ <CommonParameters>]
```

Parameter Set: ByCollectionIdAndExcludeCollectionId

```
Get-CMUserCollectionExcludeMembershipRule -CollectionId <String> -ExcludeCollectionId <String> [ <CommonParameters>]
```

Parameter Set: ByCollectionIdAndExcludeCollectionName

```
Get-CMUserCollectionExcludeMembershipRule -CollectionId <String> -ExcludeCollectionName <String> [ <CommonParameters>]
```

Parameter Set: ByCollectionNameAndExcludeCollectionId

```
Get-CMUserCollectionExcludeMembershipRule -CollectionName <String> -ExcludeCollectionId <String> [ <CommonParameters>]
```

Parameter Set: ByCollectionValueAndExcludeCollectionId

```
Get-CMUserCollectionExcludeMembershipRule -Collection <IResultObject> -ExcludeCollectionId <String> [ <CommonParameters>]
```

Parameter Set: ByCollectionValueAndExcludeCollectionName

```
Get-CMUserCollectionExcludeMembershipRule -Collection <IResultObject> -ExcludeCollectionName <String> [ <CommonParameters>]
```

Detailed Description

The **Get-CMUserCollectionExcludeMembershipRule** cmdlet retrieves the rules that exclude the members of another collection from the user collections where the rule is applied. You can specify the user collections where the rule is applied by using their names, IDs, or by specifying an object that represents the collections. You can specify the collection whose members are excluded by using its name, ID, or an object that represents the collection.

Microsoft System Center 2012 Configuration Manager dynamically updates the membership of the user collection on a schedule if the membership of the excluded collection changes. For more information

about membership rules, see [Introduction to Collections in Configuration Manager](http://go.microsoft.com/fwlink/p/?LinkID=259433) (http://go.microsoft.com/fwlink/p/?LinkID=259433) on TechNet.

Parameters

-Collection<IResultObject>

Specifies a Configuration Manager user collection object. To obtain a user collection object, use the **Get-CMUserCollection** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CollectionId<String>

Specifies the IDs of the user collections where the rule is applied.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CollectionName<String>

Specifies the name of a user collection where the rule is applied.

Aliases	none
Required?	true
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ExcludeCollectionId<String>

Specifies the ID of the collection whose members are excluded from the user collections.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ExcludeCollectionName<String>

Specifies the name of the collection whose members are excluded from the user collections.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

<CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about_CommonParameters](#).

Examples

Example 1: Get the exclude membership rule from a user collection

This command gets the rule that excludes the members of the collection that has the ID SMSDM001 from the user collection that has the ID 9990000D.

```
PS C:\> Get-CMUserCollectionExcludeMembershipRule -CollectionId "9990000D" -  
ExcludeCollectionId "SMSDM001"
```

Related topics

[Add-CMUserCollectionExcludeMembershipRule](#)

[Remove-CMUserCollectionExcludeMembershipRule](#)

[Get-CMUserCollection](#)

Get-CMUserCollectionIncludeMembershipRule

Get-CMUserCollectionIncludeMembershipRule

Gets the include membership rules from one or more user collections in the Configuration Manager hierarchy.

Syntax

Parameter Set: ByCollectionNameAndIncludeCollectionName

```
Get-CMUserCollectionIncludeMembershipRule -CollectionName <String> -IncludeCollectionName <String> [ <CommonParameters>]
```

Parameter Set: ByCollectionIdAndIncludeCollectionId

```
Get-CMUserCollectionIncludeMembershipRule -CollectionId <String> -IncludeCollectionId <String> [ <CommonParameters>]
```

Parameter Set: ByCollectionIdAndIncludeCollectionName

```
Get-CMUserCollectionIncludeMembershipRule -CollectionId <String> -IncludeCollectionName <String> [ <CommonParameters>]
```

Parameter Set: ByCollectionNameAndIncludeCollectionId

```
Get-CMUserCollectionIncludeMembershipRule -CollectionName <String> -IncludeCollectionId <String> [ <CommonParameters>]
```

Parameter Set: ByCollectionValueAndIncludeCollectionId

```
Get-CMUserCollectionIncludeMembershipRule -Collection <IResultObject> -IncludeCollectionId <String> [ <CommonParameters>]
```

Parameter Set: ByCollectionValueAndIncludeCollectionName

```
Get-CMUserCollectionIncludeMembershipRule -Collection <IResultObject> -IncludeCollectionName <String> [ <CommonParameters>]
```

Detailed Description

The **Get-CMUserCollectionIncludeMembershipRule** cmdlet retrieves rules that include the members of another collection in the user collections where the rule is applied. You can specify the user collections where the rule is applied by using their names, IDs, or by specifying an object that represents the collections.

Microsoft System Center 2012 Configuration Manager dynamically updates the membership of the user collection if the membership of the included collection changes. For more information about membership rules, see [Introduction to Collections in Configuration Manager](http://go.microsoft.com/fwlink/?LinkID=259433) (<http://go.microsoft.com/fwlink/?LinkID=259433>) on TechNet.

Parameters

-Collection<IResultObject>

Specifies a Configuration Manager user collection object where the rule is applied. To obtain a user collection object, use the **Get-CMUserCollection** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CollectionId<String>

Specifies the ID of the user collection where the rule is applied.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CollectionName<String>

Specifies the name of the user collection where the rule is applied.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-IncludeCollectionId<String>

Specifies the ID for the collection whose members are included in the user collections.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-IncludeCollectionName<String>

Specifies the name for the collection whose members are included in the user collections.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

<CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about_CommonParameters](#).

Examples

Example 1: Get the include membership rules from a user collection

This command gets the include membership rules for the collection that has the ID SMSDM001 from the user collection that has the ID 9990000D.

```
PS C:\> Get-CMUserCollectionIncludeMembershipRule -CollectionId "9990000D" -  
IncludeCollectionId "SMSDM001"
```

Related topics

[Get-CMUserCollection](#)

[Add-CMUserCollectionIncludeMembershipRule](#)

[Remove-CMUserCollectionIncludeMembershipRule](#)

Get-CMUserCollectionQueryMembershipRule

Get-CMUserCollectionQueryMembershipRule

Gets the query membership rules from one or more user collections in the Configuration Manager hierarchy.

Syntax

Parameter Set: ByCollectionName

```
Get-CMUserCollectionQueryMembershipRule -CollectionName <String> -RuleName <String> [  
<CommonParameters>]
```

Parameter Set: ByCollectionId

```
Get-CMUserCollectionQueryMembershipRule -CollectionId <String> -RuleName <String> [  
<CommonParameters>]
```

Parameter Set: ByCollectionValue

```
Get-CMUserCollectionQueryMembershipRule -Collection <IResultObject> -RuleName <String> [  
<CommonParameters>]
```

Detailed Description

The **Get-CMUserCollectionQueryMembershipRule** cmdlet retrieves rules from the specified user collections. You can specify the user collections where the rule is applied by using their names, IDs, or by specifying an input object that represents the user collections. The query is specified by its ID or name.

A query rule lets you dynamically update the membership of a collection based on a query that is run on a schedule. For more information about membership rules in Microsoft System Center 2012 Configuration Manager, see [Introduction to Collections in Configuration Manager](http://go.microsoft.com/fwlink/p/?LinkID=259433) (<http://go.microsoft.com/fwlink/p/?LinkID=259433>) on TechNet.

Parameters

-Collection<IResultObject>

Specifies a Configuration Manager user collection object. To obtain a user collection object, use the **Get-CMUserCollection** cmdlet.

Aliases	none
---------	------

Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CollectionId<String>

Specifies the ID of the user collection where the rule is applied.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CollectionName<String>

Specifies the name of the user collection where the rule is applied.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RuleName<String>

Specifies the name of the query rule.

Aliases	none
---------	------

Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

<CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about_CommonParameters](#).

Examples

Example 1: Get a rule by using a collection name

This command gets the rule named Remote Users by Domain that belongs to the collection named Remote Users.

```
PS C:\> Get-CMUserCollectionQueryMembershipRule -CollectionName "Remote Users" -RuleName "Remote Users by Domain"
```

Related topics

[Get-CMUserCollection](#)

[Add-CMUserCollectionQueryMembershipRule](#)

[Remove-CMUserCollectionQueryMembershipRule](#)

Get-CMUserDataAndProfileConfigurationItem

Get-CMUserDataAndProfileConfigurationItem

Gets user data and profiles configuration items.

Syntax

Parameter Set: SearchByIdMandatory

```
Get-CMUserDataAndProfileConfigurationItem -Id <String[]> [ <CommonParameters>]
```

Parameter Set: SearchByName

```
Get-CMUserDataAndProfileConfigurationItem [-Name <String[]> ] [ <CommonParameters>]
```

Detailed Description

The **Get-CMUserDataAndProfileConfigurationItem** cmdlet gets one or more user data and profiles configuration items in Microsoft System Center 2012 Configuration Manager.

Parameters

-Id<String[]>

Specifies an array of IDs of user data and profiles configuration items.

Aliases	CId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String[]>

Specifies an array of names of user data and profiles configuration items.

Aliases	LocalizedDisplayName
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

<CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about_CommonParameters](#).

Examples

Example 1: Get a user data and profiles configuration item

The command gets the user data and profiles configuration item that is named UDPConfig02.

```
PS C:\> Get-CMUserDataAndProfileConfigurationItem -Name "UDPConfig02"
```

Related topics

[Copy-CMUserDataAndProfileConfigurationItem](#)

[New-CMUserDataAndProfileConfigurationItem](#)

[Remove-CMUserDataAndProfileConfigurationItem](#)

[Set-CMUserDataAndProfileConfigurationItem](#)

Get- CMUserDataAndProfileConfigurationItemXmlDe finition

Get-**CMUserDataAndProfileConfigurationItemXmlDefinition**

Gets the XML definition file for user data and profile configuration items.

Syntax

Parameter Set: SearchByNameMandatory

```
Get-CMUserDataAndProfileConfigurationItemXmlDefinition -Name <String[]> [  
<CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Get-CMUserDataAndProfileConfigurationItemXmlDefinition -Id <String[]> [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Get-CMUserDataAndProfileConfigurationItemXmlDefinition -InputObject <IResultObject> [  
<CommonParameters>]
```

Detailed Description

The **Get-**CMUserDataAndProfileConfigurationItemXmlDefinition**** cmdlet gets the XML definition file for user data and profile configuration items in Microsoft System Center 2012 Configuration Manager. This information can be useful when you want to author configuration data manually.

Parameters

-Id<String[]>

Specifies an array of IDs of user data and profile configuration items.

Aliases	Ciid
Required?	true
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a **CMUserDataAndProfileConfigurationItem** object. To obtain a **CMUserDataAndProfileConfigurationItem** object, use the **Get-
CMUserDataAndProfileConfigurationItem** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String[]>

Specifies an array of names of user data and profile configuration items.

Aliases	LocalizedDisplayName
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

<CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -
OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

Examples

Example 1: Get the XML definition file for a user data and profile configuration item

This command gets the XML definition file for the user data and profile configuration item named UDPConfig02.

```
PS C:\> Get-CMUserDataAndProfileConfigurationItemXmlDefinition -Name "UDPConfig02"
```

Related topics

[Get-CMUserDataAndProfileConfigurationItem](#)

[Copy-CMUserDataAndProfileConfigurationItem](#)

[Remove-CMUserDataAndProfileConfigurationItem](#)

Get-CMUserDeviceAffinity

Get-CMUserDeviceAffinity

Gets user device affinities.

Syntax

Parameter Set: SearchByDeviceIdMandatory

```
Get-CMUserDeviceAffinity -DeviceId <String[]> [ <CommonParameters>]
```

Parameter Set: SearchByDeviceNameMandatory

```
Get-CMUserDeviceAffinity -DeviceName <String[]> [ <CommonParameters>]
```

Parameter Set: SearchByUserIdMandatory

```
Get-CMUserDeviceAffinity -UserId <String[]> [ <CommonParameters>]
```

Parameter Set: SearchByUsernameMandatory

```
Get-CMUserDeviceAffinity -UserName <String[]> [ <CommonParameters>]
```

Detailed Description

The **Get-CMUserDeviceAffinity** cmdlet gets one or more user device affinities. User device affinity in Microsoft System Center 2012 Configuration Manager is a method of associating a user with one or more specified devices.

Parameters

-DeviceId<String[]>

Specifies an array of device IDs.

Aliases	ResourceId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeviceName<String[]>

Specifies an array of device names.

Aliases	ResourceName
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UserId<String[]>

Specifies an array of IDs of the primary users of the devices.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UserName<String[]>

Specifies an array of names of the primary users of the devices.

Aliases	UniqueUserName
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

<CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

Examples

Example 1: Get a user device affinity by using a user name

This command gets the user device affinity for the user named CENTRAL\001D\$.

```
PS C:\> Get-CMUserDeviceAffinity -UserName "CENTRAL\001D$"
```

Example 2: Get a user device affinity by using a user ID

This command gets the user device affinity for the user that has the ID named 2063597981.

```
PS C:\> Get-CMUserDeviceAffinity -UserID "2063597981"
```

Example 3: Get a user device affinity by using a device name

This command gets the user device affinity for the device named CMCEN-DIST02.

```
PS C:\> Get-CMUserDeviceAffinity -DeviceName "CMCEN-DIST02"
```

Example 4: Get a user device affinity by using a device ID

This command gets the user device affinity for the device that has the ID 2097152000.

```
PS C:\> Get-CMUserDeviceAffinity -DeviceID "2097152000"
```

Related topics

[Import-CMUserDeviceAffinity](#)

[Get-CMUserDeviceAffinityRequest](#)

[Approve-CMUserDeviceAffinityRequest](#)

[Deny-CMUserDeviceAffinityRequest](#)

Get-CMUserDeviceAffinityRequest

Get-CMUserDeviceAffinityRequest

Gets a request for user device affinity in Configuration Manager.

Syntax

Parameter Set: SearchByNameMandatory

```
Get-CMUserDeviceAffinityRequest -CollectionName <String> [-DeviceId <String> ] [-DeviceName <String> ] [-UserId <String> ] [-UserName <String> ] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Get-CMUserDeviceAffinityRequest -CollectionId <String> [-DeviceId <String> ] [-DeviceName <String> ] [-UserId <String> ] [-UserName <String> ] [ <CommonParameters>]
```

Detailed Description

The **Get-CMUserDeviceAffinityRequest** cmdlet gets a request for user device affinity.

In Microsoft System Center 2012 Configuration Manager, user device affinity defines a relationship between a user and a device. Instead of deploying an application to a group of devices, you deploy an application to a user and Configuration Manager installs the application on all devices associated with the user.

Parameters

-CollectionId<String>

Specifies a collection ID that represents the user device affinity in Configuration Manager.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CollectionName<String>

Specifies a name of a collection that represents the user device affinity in Configuration Manager.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeviceId<String>

Specifies a device ID in Configuration Manager.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeviceName<String>

Specifies a device name in Configuration Manager.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UserId<String>

Specifies a Configuration Manager ID for a user resource.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UserName<String>

Specifies a user name for a resource in Configuration Manager.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

<CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

Examples

Example 1: Get a request for user device affinity

This command gets a request for user device affinity for the collection named All Systems.

```
PS C:\> Get-CMUserDeviceAffinityRequest -CollectionName "All Systems"
```

Related topics

[Approve-CMUserDeviceAffinityRequest](#)

[Deny-CMUserDeviceAffinityRequest](#)

Get-CMVhd

Get-CMVhd

Gets VHD images.

Syntax

Parameter Set: SearchByName

```
Get-CMVhd [-Name <String> ] [-SecuredScopeNames <String> ] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Get-CMVhd -Id <String[]> [-SecuredScopeNames <String> ] [ <CommonParameters>]
```

Detailed Description

The **Get-CMVhd** cmdlet gets one or more virtual hard disk (VHD) images that were created by using the operating system deployment feature.

Parameters

-Id<String[]>

Specifies an array of IDs of VHD images.

Aliases	Packageld
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies the name of a VHD image.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SecuredScopeNames<String>

Specifies names of security scopes. A security scope name can be Default or the name of a custom-created security scope.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

<CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about_CommonParameters](#).

Examples

Example 1: Get a VHD image by using its name

This command gets the VHD image named VHDAccounting08.

```
PS C:\> Get-CMVhd -Name "VHDAccounting08"
```

Related topics

[New-CMVhd](#)

[Remove-CMVhd](#)

[Set-CMVhd](#)



Get-CMVpnProfileConfigurationItem

Get-CMVpnProfileConfigurationItem

Gets a VPN profile.

Syntax

Parameter Set: SearchByName

```
Get-CMVpnProfileConfigurationItem [-Name <String[]> ] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Get-CMVpnProfileConfigurationItem -Id <String[]> [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Get-CMVpnProfileConfigurationItem -InputObject <IResultObject> [ <CommonParameters>]
```

Detailed Description

The **Get-CMVpnProfileConfigurationItem** cmdlet gets a virtual private network (VPN) profile.

Parameters

-Id<String[]>

Specifies an array of IDs of VPN profile objects.

Aliases	CId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a VPN profile object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String[]>

Specifies an array of names of VPN profiles.

Aliases	LocalizedDisplayName
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

<CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

Examples

Example 1: Get a VPN profile

This command gets the VPN profile that has the ID 16777568.

```
PS C:\> Get-CMVpnProfileConfigurationItem -Id "16777568"
```

Related topics

[Copy-CMVpnProfileConfigurationItem](#)

[New-CMVpnProfileConfigurationItem](#)

[Remove-CMVpnProfileConfigurationItem](#)

[Set-CMVpnProfileConfigurationItem](#)

Get-CMWindowsFirewallPolicy

Get-CMWindowsFirewallPolicy

Gets Windows Firewall policies for Endpoint Protection.

Syntax

Parameter Set: SearchByName

```
Get-CMWindowsFirewallPolicy [-Name <String[]> ] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Get-CMWindowsFirewallPolicy -Id <String[]> [ <CommonParameters>]
```

Detailed Description

The **Get-CMWindowsFirewallPolicy** cmdlet gets one or more Windows Firewall policies for System Center 2012 Endpoint Protection in Microsoft System Center 2012 Configuration Manager.

Parameters

-Id<String[]>

Specifies an array of IDs of Windows Firewall policies.

Aliases	CId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String[]>

Specifies an array of names of Windows Firewall policies.

Aliases	LocalizedDisplayName
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

<CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about_CommonParameters](#).

Examples

Example 1: Get a Windows Firewall policy by using an ID

This command gets the Windows Firewall policy that has the ID 16777568.

```
PS C:\> Get-CMWindowsFirewallPolicy -Id "16777568"
```

Example 2: Get a Windows Firewall policy by using a name

This command gets the Windows Firewall policy that has the name WFPContoso01.

```
PS C:\> Get-CMWindowsFirewallPolicy -Name "WFPContoso01"
```

Related topics

[Set-CMWindowsFirewallPolicy](#)

[New-CMWindowsFirewallPolicy](#)

[Remove-CMWindowsFirewallPolicy](#)

Get-CMWirelessProfileConfigurationItem

Get-CMWirelessProfileConfigurationItem

Gets wireless profiles.

Syntax

Parameter Set: SearchByName

```
Get-CMWirelessProfileConfigurationItem [-Name <String[]> ] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Get-CMWirelessProfileConfigurationItem -Id <String[]> [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Get-CMWirelessProfileConfigurationItem -InputObject <IResultObject> [ <CommonParameters>]
```

Detailed Description

The **Get-CMWirelessProfileConfigurationItem** cmdlet gets wireless profiles.

Parameters

-Id<String[]>

Specifies an array of IDs of wireless profiles.

Aliases	CId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a wireless profile object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String[]>

Specifies an array of names of wireless profiles.

Aliases	LocalizedDisplayName
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

<CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

Examples

Example 1: Get a wireless profile

This command gets the wireless profile named AP_Test.

```
PS C:\> Get-NetWirelessProfileConfigurationItem -Name "AP_Test"
```

Related topics

[Copy-CMWirelessProfileConfigurationItem](#)

[New-CMWirelessProfileConfigurationItem](#)

[Remove-CMWirelessProfileConfigurationItem](#)

[Set-CMWirelessProfileConfigurationItem](#)

Import-CMAntimalwarePolicy

Import-CMAntimalwarePolicy

Imports an antimalware policy for Endpoint Protection.

Syntax

Parameter Set: Import

```
Import-CMAntimalwarePolicy -ImportFilePath <String> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Detailed Description

The **Import-CMAntimalwarePolicy** cmdlet imports antimalware policies for System Center 2012 Endpoint Protection. When you enable Endpoint Protection, Microsoft System Center 2012 Configuration Manager applies a default antimalware policy to client computers. System Center 2012 Configuration Manager supplies a selection of predefined antimalware policy templates. You can import into these predefined templates into System Center 2012 Configuration Manager and assign them to System Center 2012 Configuration Manager client computers in your hierarchy. These templates are available in the folder *<ConfigMgr Install Folder>\AdminConsole\XMLStorage\EPTemplates*.

Parameters

-ImportFilePath<String>

Specifies the Universal Naming Convention (UNC) path of the policy file to import.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Import an antimalware policy

This command imports the antimalware policy from the file named SCEP12_Standard_Desktop.xml.

```
PS C:\> Import-CMAntimalwarePolicy -ImportFilePath  
"\\Contosofs01\Mid\SCEP12_Standard_Desktop.xml"
```

Related topics

[Get-CMAntiMalwarePolicy](#)

[Set-CMAntiMalwarePolicy](#)

[New-CMAntimalwarePolicy](#)

[Remove-CMAntiMalwarePolicy](#)

[Export-CMAntimalwarePolicy](#)

[Merge-CMAntimalwarePolicy](#)

Import-CMApplication

Import-CMApplication

Imports an application in Configuration Manager.

Syntax

Parameter Set: Default

```
Import-CMApplication -FilePath <String> [-ImportActionType {NotSet | Skip | DirectImport |  
Rename | Overwrite | ImportFail} ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Import-CMApplication** cmdlet imports a package created by the **Export-CMApplication** cmdlet. A package contains one or more applications and related objects, such as catalogs. If the package contains content, the application package imports the content, or includes a reference to the content.

Parameters

-FilePath<String>

Specifies a file path for the application.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ImportActionType<ImportActionType>

Specifies an import action type for the application. Valid values are:

- ImportFail. Application and dependencies will not be imported.
- NotImport. Map objects due to exact match of scope, version, or name.

- DirectImport. Import objects.
- Overwrite. Map objects despite name or scope duplication.
- Rename. Create new.

The acceptable values for this parameter are:

NotSet	
Skip	
DirectImport	
Rename	
Overwrite	
ImportFail	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Import an application by using a file path

This command imports an application by using the specified file path.

```
PS C:\> Import-CMApplication -FilePath "\\Contoso01\CM\Toolbox\Test.zip"
```

Related topics

[Export-CMApplication](#)

[Get-CMApplication](#)

[New-CMApplication](#)

[Remove-CMApplication](#)

[Resume-CMApplication](#)

[Set-CMApplication](#)

[Suspend-CMApplication](#)

Import-CMBaseline

Import-CMBaseline

Imports Configuration Manager baselines.

Syntax

Parameter Set: ImportByFileName

```
Import-CMBaseline -FileName <String[]> [-DuplicateWhileImporting] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Detailed Description

The **Import-CMBaseline** cmdlet imports Microsoft System Center 2012 Configuration Manager baselines from files. A baseline is a collection of configuration items that System Center 2012 Configuration Manager uses to evaluate whether a computer complies with software requirements. After you import a baseline, you can deploy it to a collection so that devices in that collection download the configuration baseline and assess compliance with it.

You can import a configuration baseline from a .cab file that conforms to the Service Modeling Language (SML) schema. For example, you might import data previously exported from System Center 2012 Configuration Manager or best practices included in a Monitoring Pack for Configuration Manager.

When you import a baseline configuration, you have the option of creating a local copy. You can modify that baseline in the future.

Parameters

-DuplicateWhileImporting

Indicates that the cmdlet duplicates a baseline while it imports the baseline. If you duplicate a baseline, you can modify that baseline in the future.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-FileName<String[]>

Specifies an array of .cab file names. Each file contains Configuration Manager configuration items.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Import a baseline

This command imports a baseline from a file named BaselineW2K8.cab.

```
PS C:\> Import-CMBaseline -FileName "\\ContosoServer01\Public\CM\BaselineW2K8.cab"
```

Example 2: Import multiple baselines

This command imports baselines from .cab files named BaselineW2K8.cab and BaselineWin7.cab. This command uses the *DuplicateWhileImporting* parameter, so the command creates an editable version of the configuration baselines.

```
PS C:\> Import-CMBaseline -FileName  
"\\ContosoServer01\Public\CM\BaselineW2K8.cab", "\\ContosoServer01\Public\CM\BaselineWin7.cab  
" -DuplicateWhileImporting
```

Related topics

[Disable-CMBaseline](#)

[Enable-CMBaseline](#)

[Export-CMBaseline](#)

[Get-CMBaseline](#)

[New-CMBaseline](#)

[Remove-CMBaseline](#)

[Set-CMBaseline](#)

Import-CMCertificate

Import-CMCertificate

Imports a certificate.

Syntax

Parameter Set: Default

```
Import-CMCertificate -Path <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Import-CMCertificate** cmdlet imports a public key infrastructure (PKI) certificate to Microsoft System Center 2012 Configuration Manager.

Parameters

-Path<String>

Specifies a certification path.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Import a certificate

This command imports the PKI certificate from the file named BaseCert.

```
PS C:\> Import-CMCertificate -Path "\\Contoso01\CM\Certificates\BaseCert.txt"
```

Related topics

[Update-CMCertificate](#)

[Block-CMCertificate](#)

[Unblock-CMCertificate](#)

Import-CMComputerInformation

Import-CMComputerInformation

Imports computer information into a Configuration Manager database.

Syntax

Parameter Set: ImportComputerByUsingFile

```
Import-CMComputerInformation -CollectionName <String> -FileName <String> [-  
EnableColumnHeadings <Boolean> ] [-VariableName <String> ] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: ImportSingleComputer

```
Import-CMComputerInformation -CollectionName <String> -ComputerName <String> -MacAddress  
<String> [-SmBiosGuid <String> ] [-SourceComputerName <String> ] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Detailed Description

The **Import-CMComputerInformation** cmdlet imports computer information directly into a Microsoft System Center 2012 Configuration Manager database. For System Center 2012 Configuration Manager to deploy an operating system to a new computer with no installed operating system, you must add the new computer to System Center 2012 Configuration Manager. After you import the computer information, System Center 2012 Configuration Manager can deploy an operating system.

You can import a single computer by specifying the Media Access Control (MAC) address and computer name, along with the name of a collection. This cmdlet adds this computer to the specified collection.

You can also import several computers by specifying a Comma Separated Values .csv file with computer information, along with the name of a collection. This cmdlets adds the computers to the specified collection.

You can specify the name of a reference computer. System Center 2012 Configuration Manager migrates user information and settings from the reference computer to the new computer.

Parameters

-CollectionName<String>

Specifies a name of a Configuration Manager device collection.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ComputerName<String>

Specifies a name of a computer.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnableColumnHeadings<Boolean>

Specifies that the computer information file contains a header row. If you enter a value of \$True, the cmdlet ignores the first row of the file.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-FileName<String>

Specifies a .csv file that contains computer information. The file must contain the name and MAC address of each computer to be imported.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-MacAddress<String>

Specifies a MAC address for a computer in the format (00:00:00:00:00:00). The Windows Preinstallation Environment (Windows PE) must have a driver for the specified network adapter.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SmbiosGuid<String>

Specifies a GUID for the system management BIOS (SMBIOS) of a computer.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SourceComputerName<String>

Specifies a name of a reference computer. Configuration Manager migrates user state and settings from the reference computer to the new computer.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-VariableName<String>

Specifies a variable name for an imported column. When you import a .csv file, you specify the columns to import and assign them to a Configuration Manager field. A variable allows you to assign a column to a variable.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Import computers by using a file

This command imports the computers specified in the CVS file into the All Systems collection. This command includes a value of `$True` for the `-EnableColumnHeadings` parameter. The cmdlet ignores the first line of the file.

```
PS C:\> Import-CMComputerInformation -CollectionName "All Systems" -FileName
"\\cshare\Public\CM\ImportComputers.csv" -EnableColumnHeadings $True
```

Example 2: Import a single computer

This command imports a specified computer into the All Systems collection. The command specifies the name, MAC address, and SMBIOS GUID for a computer.

```
PS C:\> Import-CMComputerInformation -CollectionName "All Systems" -ComputerName
"Computer08" -MacAddress "5F:DA:FA:FA:FA:FA" -SmBiosGuid "AAAAAAAA-AAAA-AAAA-AAAA-
AAAAAAAAAAAA"
```

Example 3: Import a computer using a reference computer

This command imports a specified computer into the All Systems collection. The command specifies the name, MAC address, and SMBIOS GUID for a computer. The command also includes a reference computer to associate with the new computer.

```
PS C:\> Import-CMComputerInformation -CollectionName "All Systems" -ComputerName  
"Computer08" -MacAddress "5F:DA:FA:FA:FA:FA" -SmBiosGuid "AAAAAAAA-AAAA-AAAA-AAAA-  
AAAAAAAAAAAA" -SourceComputerName "ResourceComputer01"
```

Related topics

[Get-CMComputerAssociation](#)

Import-CMConfigurationItem

Import-CMConfigurationItem

Imports Configuration Manager configuration items.

Syntax

Parameter Set: ImportByFileName

```
Import-CMConfigurationItem -FileName <String[]> [-DuplicateWhileImporting] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Import-CMConfigurationItem** cmdlet imports Microsoft System Center 2012 Configuration Manager configuration items from one or more cabinet files. The files that you import must conform to the Service Modeling Language (SML) schema and can contain information about configuration data from one of the following sources:

- Best practices from a System Center 2012 Configuration Manager Configuration Pack.
- Configuration data that you have externally authored and packaged into a cabinet (.cab) file.
- Configuration data exported from System Center 2012 Configuration Manager.

Configuration items contain one or more settings, along with compliance rules. Items usually define a unit of configuration you want to monitor.

Parameters

-DuplicateWhileImporting

Indicates that Configuration Manager imports configuration items that exist in Configuration Manager as duplicate configuration items.

Use this parameter to create a configuration item when you want an exact copy of an configuration item that you import to use as your starting point, but you want to modify it to create an independent configuration item from the original.

Aliases	none
Required?	false
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-FileName<String[]>

Specifies an array of names of cabinet (cab) files.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Import configuration items

This command imports configuration items from the files 7389_OSCI.cab and 7452OS_1.cab.

```
PS C:\> Import-CMConfigurationItem -FileName "\\atc-dist-01\Public\CM\AdminUITeam\CIData\7389_OSCI.cab", "\\atc-dist-01\Public\CM\AdminUITeam\CIData\7452OS_1.cab"
```

Example 2: Import configuration items and create duplicate configuration items

This command imports configuration items from the files 7389_OSCI.cab and 7452OS_1.cab. The *DuplicateWhileImporting* parameter indicates that imports configuration items that exist in System Center 2012 Configuration Manager as duplicate configuration items.

```
PS C:\> Import-CMConfigurationItem -FileName "\\Contoso01\Public\CM\7389_OSCI.cab", "\\Contoso01\Public\CM\7452OS_1.cab" -DuplicateWhileImporting
```

Related topics

[Get-CMConfigurationItem](#)

[Set-CMConfigurationItem](#)

[New-CMConfigurationItem](#)

[Remove-CMConfigurationItem](#)

[Export-CMConfigurationItem](#)

Import-CMDeviceCollection

Import-CMDeviceCollection

Imports a device collection.

Syntax

Parameter Set: SearchByNameMandatory

```
Import-CMDeviceCollection -ImportFilePath <String> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Detailed Description

The **Import-CMDeviceCollection** cmdlet imports a device collection. This cmdlet imports a device collection into the site database from a Managed Object Format (MOF) file.

Parameters

-ImportFilePath<String>

Specifies the full path of the import file.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Import a device collection into the site database

This command imports a device collection into the site database from the MOF file named FinancePCs.

```
PS C:\> Import-CMDeviceCollection -ImportFilePath "c:\assets\FinancePCs.mof"
```

Related topics

[Export-CMDeviceCollection](#)

[Get-CMDeviceCollection](#)

[New-CMDeviceCollection](#)

[Remove-CMDeviceCollection](#)

[Set-CMDeviceCollection](#)

Import-CMDriver

Import-CMDriver

Imports device drivers into the driver catalog.

Syntax

Parameter Set: NewDriver

```
Import-CMDriver -UncFileLocation <String> [-AdministrativeCategory <IResultObject[]> ] [-  
BootImagePackage <IResultObject[]> ] [-DriverPackage <IResultObject[]> ] [-  
EnableAndAllowInstall <Boolean> ] [-ImportDuplicateDriverOption {AppendCategory |  
KeepExistingCategory | NotImport | OverwriteCategory} ] [-SupportedPlatformName <String[]> ]  
[-UpdateDistributionPointsforBootImagePackage <Boolean> ] [-  
UpdateDistributionPointsforDriverPackage <Boolean> ] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Detailed Description

The **Import-CMDriver** cmdlet imports one or more device drivers into the driver catalog.

As part of the import process for the device driver, Microsoft System Center 2012 Configuration Manager reads the provider, class, version, signature, supported hardware, and supported platform information that is associated with the device. By default, the driver is named after the first hardware device that it supports; however, you can rename the device driver later. The supported platforms list is based on the information in the INF file of the driver. Because the accuracy of this information can vary, manually verify that the device driver is supported after it is imported into the driver catalog.

When you import device drivers into the catalog, you can add the device drivers to driver packages or to boot image packages.

Parameters

-AdministrativeCategory<IResultObject[]>

Specifies an array of administrative categories. Assign the device drivers to an administrative category for filtering purposes, such as Desktops or Notebooks categories.

To obtain an administrative category object, use the **Get-CMCategory** cmdlet.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-BootImagePackage<IResultObject[]>

Specifies an array of boot image objects. Use this parameter to specify the boot images that can install the imported device drivers. To obtain a boot image object, use the **Get-CMBootImage** cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DriverPackage<IResultObject[]>

Specifies an array of driver package objects. Use this parameter to specify the driver packages that Configuration Manager uses to distribute the device drivers. To obtain a driver package object, use the **Get-CMDriverPackage** cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnableAndAllowInstall<Boolean>

Indicates whether Configuration Manager enables the drivers and allows computers to install the drivers.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ImportDuplicateDriverOption<ImportDuplicateDriverOption>

Specifies how Configuration Manager manages driver categories when you import a duplicate device driver. Valid values are:

- AppendCategory
- KeepExistingCategory
- NotImport
- OverwriteCategory

The acceptable values for this parameter are:

AppendCategory	
KeepExistingCategory	
NotImport	
OverwriteCategory	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SupportedPlatformName<String[]>

Specifies an array of names of platforms on which the device driver can run.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UncFileLocation<String>

Specifies the network path (UNC) of the device drivers.

To import all the device drivers that are contained in a specific folder, specify the network path to the device driver folder. For example: \\servername\folder. To import a specific driver from a folder, specify the network path (UNC) to the Windows device driver .INF or mass storage Txtsetup.oem file of the driver.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UpdateDistributionPointsforBootImagePackage<Boolean>

Indicates whether Configuration Manager updates distribution points when the device drivers are added to the boot image package.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UpdateDistributionPointsforDriverPackage<Boolean>

Indicates whether Configuration Manager updates distribution points when the device drivers are added to the driver package.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Import a device driver

The first command uses the **Get-CMDriverPackage** cmdlet to get the driver package that has the ID ST100062, and stores the result in the \$driverPackage variable.

The second command uses the **Get-CMBootImage** cmdlet to get the boot image that has the ID CM100004, and stores the result in the \$bootPackage variable.

The third command imports the Windows device driver named smwdmCH6.inf. The command specifies that Configuration Manager overwrites the driver category when you import a duplicate device drive. The command enables the drivers and allows computers to install the drivers. The command specifies that Configuration Manager use the driver packages stored in \$driverPackage to distribute the device drivers, and specifies the boot images stored in \$bootPackage that can install the imported device drivers.

```
PS C:\> $driverPackage = Get-CMDriverPackage -Id "ST100062"
PS C:\> $bootPackage = Get-CMBootImage -Id "CM100004"
PS C:\> Import-CMDriver -UncFileLocation "\\btc-dist-08\Public\CM\AdminTeam\Driver\X64Driver\AudioDriver\smwdmCH6.inf" -
ImportDuplicateDriverOption OverwriteCategory -EnableAndAllowInstall $True -DriverPackage
$driverPackage -BootImagePackage $bootPackage
```

Related topics

[Get-CMDriver](#)

[Set-CMDriver](#)

[Enable-CMDriver](#)

[Disable-CMDriver](#)

[Remove-CMDriver](#)

[Get-CMCategory](#)

[Get-CMBootImage](#)

[Get-CMDriverPackage](#)

Import-CMDriverPackage

Import-CMDriverPackage

Imports a driver package.

Syntax

Parameter Set: Default

```
Import-CMDriverPackage -ImportFilePath <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Import-CMDriverPackage** cmdlet imports a driver packages to Microsoft System Center 2012 Configuration Manager. You can use the [Export-CMDriverPackage](#) cmdlet to export a driver package to a .zip file.

Parameters

-ImportFilePath<String>

Specifies the Universal Naming Convention (UNC) path of the import file.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Import a driver package

This command imports a driver package from the import file named DriverPackage.zip.

```
PS C:\> Import-CMDriverPackage -ImportFilePath  
"\\Contoso02\main\driverpackages\DriverPackage.zip"
```

Related topics

[Export-CMDriverPackage](#)

[Get-CMDriverPackage](#)

[New-CMDriverPackage](#)

[Remove-CMDriverPackage](#)

[Set-CMDriverPackage](#)

Import-CMPackage

Import-CMPackage

Imports a Configuration Manager package.

Syntax

Parameter Set: Default

```
Import-CMPackage -ImportFilePath <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Import-CMPackage** cmdlet imports a Microsoft System Center 2012 Configuration Manager package. You can use this cmdlet to import a package of collections, queries, or reports so that you can later deploy these items to a different location.

Parameters

-ImportFilePath<String>

Specifies the Universal Naming Convention (UNC) path for the file package that contains the package that you import. The cmdlet imports all packages that the file package contains.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Import a package

This command imports a package from the path \\Deploy01\ExportPackages.

```
PS C:\> Import-CMPackage -ImportFilePath "\\Deploy01\ExportPackages"
```

Related topics

[Export-CMPackage](#)

[Get-CMPackage](#)

[New-CMPackage](#)

[Remove-CMPackage](#)

[Set-CMPackage](#)

Import-CMSecurityRole

Import-CMSecurityRole

Imports a security role into Configuration Manager.

Syntax

Parameter Set: Default

```
Import-CMSecurityRole -Overwrite <Boolean> -XmlFileName <String> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Detailed Description

The **Import-CMSecurityRole** cmdlet imports a security role that was exported from another Microsoft System Center 2012 Configuration Manager hierarchy.

Parameters

-Overwrite<Boolean>

Indicates whether the security role that you import overwrites an existing security role with the same name that you specify in the **XmlFileName** parameter.

Aliases	OverwrittenExisted
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-XmlFileName<String>

Specifies the XML export file that contains the security role definition.

Aliases	RolesXml
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Import a security role

This command imports a security role into Configuration Manager from the XML export file named RemoteAdminSecurity. The command specifies that the security role that you import overwrites an existing security role with the same name.

```
PS C:\> Import-CMSecurityRole -Overwrite $True -XmlFileName "RemoteAdminSecurity.xml"
```

Related topics

[Get-CMSecurityRole](#)

[Copy-CMSecurityRole](#)

[Set-CMSecurityRole](#)

[Remove-CMSecurityRole](#)

[Export-CMSecurityRole](#)

[Remove-CMSecurityRoleFromAdministrativeUser](#)

Import-CMSoftwareLicense

Import-CMSoftwareLicense

Imports a software license.

Syntax

Parameter Set: ImportSoftwareLicense

```
Import-CMSoftwareLicense -ImportType {GeneralLicenseStatement |  
MicrosoftVolumeLicenseStatement} -MlsFilePath <String> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Detailed Description

The **Import-CMSoftwareLicense** cmdlet imports Microsoft and non-Microsoft licensing information into the Asset Intelligence catalog in Microsoft System Center 2012 Configuration Manager.

Parameters

-ImportType<ImportType>

Specifies an import type for the software license. Valid values are:

-- GeneralLicenseStatement

-- MicrosoftVolumeLicenseStatement

A general license statement contains information about the purchased licenses for any publisher. A Microsoft Volume License Statement (MVLS) license statement contains information about the license entitlements, or number of purchased licenses, for Microsoft products.

The acceptable values for this parameter are:

GeneralLicenseStatement	
MicrosoftVolumeLicenseStatement	

Aliases	none
Required?	true

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-MlsFilePath<String>

Specifies the Universal Naming Convention (UNC) path of a valid XML-formatted licensing information file.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Import a software license

This command imports a MVLS license statement from the licensing information file named SWLicense01.xml.

```
PS C:\> Import-CMSoftwareLicense -MlsFilePath "\\Contosofs01\Mid\SWLicense01.xml" -
ImportType MicrosoftVolumeLicenseStatement
```

Import-CMTaskSequence

Import-CMTaskSequence

Imports a Configuration Manager task sequence.

Syntax

Parameter Set: Default

```
Import-CMTaskSequence -ImportFilePath <String> [-IgnoreDependency] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Detailed Description

The **Import-CMTaskSequence** cmdlet imports a Microsoft System Center 2012 Configuration Manager task sequence.

Parameters

-IgnoreDependency

Indicates that the import process ignores dependencies in the task sequence.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ImportFilePath<String>

Specifies a path to the import file in Configuration Manager. To create a file to import, use the **Export-CMTaskSequence** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Import a task sequence from a file

This command imports a task sequence by using the file named TaskSequence.zip.

```
PS C:\> Import-CMTaskSequence -ImportFilePath "\\Contoso-01\Users\AimeeLott\Desktop\TaskSequence.zip"
```

Related topics

[Disable-CMTaskSequence](#)

[Enable-CMTaskSequence](#)

[Export-CMTaskSequence](#)

[Get-CMTaskSequence](#)

[New-CMTaskSequence](#)

[Remove-CMTaskSequence](#)

[Set-CMTaskSequence](#)

Import-CMUserCollection

Import-CMUserCollection

Imports a user collection in Configuration Manager.

Syntax

Parameter Set: SearchByNameMandatory

```
Import-CMUserCollection -ImportFilePath <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Import-CMUserCollection** cmdlet imports an object that represents a user collection in Microsoft System Center 2012 Configuration Manager. To import objects, you must have Create rights for the objects.

Parameters

-ImportFilePath<String>

Specifies a path to the import file.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Import a user collection

This command imports a user collection by using a file import path.

```
PS C:\> Import-CMUserCollection -ImportFilePath "C:\Imports"
```

Related topics

[Export-CMUserCollection](#)

[Get-CMUserCollection](#)

[New-CMUserCollection](#)

[Remove-CMUserCollection](#)

[Set-CMUserCollection](#)

Import-CMUserDeviceAffinity

Import-CMUserDeviceAffinity

Imports a file that contains user and device affinities to Configuration Manager.

Syntax

Parameter Set: ImportByFileName

```
Import-CMUserDeviceAffinity -FileName <String> [-EnableColumnHeadings <Boolean> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Import-CMUserDeviceAffinity** cmdlet imports a file that contains user and device affinities to Microsoft System Center 2012 Configuration Manager. User device affinity in System Center 2012 Configuration Manager is a method of associating a user with one or more specified devices.

The devices listed in the file that you specify in the *FileName* parameter must already exist as resources in the Configuration Manager database. If they do not exist, the import will fail.

Parameters

-EnableColumnHeadings<Boolean>

Indicates that the import file has column headings for reference purposes. If the comma-separated values file that you specify in the *FileName* parameter has a header line, specify this parameter and Configuration Manager ignores the header line during the import.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-FileName<String>

Specifies a comma-separated values (.csv) file that contains a list of users and devices you want to create an affinity between. Each user and device pair must be on a separate line separated by a comma. Use the format <Domain>\<user name>,<device NetBIOS name>.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Import a user device affinity

This command imports the user device affinity in the file named Remote_Users.csv. The *EnableColumnHeadings* parameter specifies that the import file has column headings for reference purposes.

```
PS C:\> Import-CMUserDeviceAffinity -FileName "Remote_Users.csv" -EnableColumnHeadings $True
```

Related topics

[Get-CMUserDeviceAffinity](#)

Install-CMClient

Install-CMClient

Installs a Configuration Manager client.

Syntax

Parameter Set: SearchByNameMandatory

```
Install-CMClient -AlwaysInstallClient <Boolean> -ForceReinstall <Boolean> -  
IncludeDomainController <Boolean> -Name <String> -SiteCode <String> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByDeviceIdMandatory

```
Install-CMClient -AlwaysInstallClient <Boolean> -DeviceId <String> -ForceReinstall <Boolean>  
-IncludeDomainController <Boolean> -SiteCode <String> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByDeviceNameMandatory

```
Install-CMClient -AlwaysInstallClient <Boolean> -DeviceName <String> -ForceReinstall  
<Boolean> -IncludeDomainController <Boolean> -SiteCode <String> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByDeviceValueMandatory

```
Install-CMClient -AlwaysInstallClient <Boolean> -Device <IResultObject> -ForceReinstall  
<Boolean> -IncludeDomainController <Boolean> -SiteCode <String> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Install-CMClient -AlwaysInstallClient <Boolean> -CollectionId <String> -ForceReinstall  
<Boolean> -IncludeDomainController <Boolean> -SiteCode <String> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Install-CMClient -AlwaysInstallClient <Boolean> -ForceReinstall <Boolean> -  
IncludeDomainController <Boolean> -InputObject <IResultObject> -SiteCode <String> [-Confirm]  
[-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Install-CMClient** cmdlet installs a client for Microsoft System Center 2012 Configuration Manager.

Parameters

-AlwaysInstallClient<Boolean>

Indicates whether Configuration Manager always installs the client.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CollectionId<String>

Specifies the ID of the collection to which the client belongs.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Device<IResultObject>

Specifies a Configuration Manager device object. You can get a Configuration Manager device object by using the **Get-CMDevice** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-DeviceId<String>

Specifies the ID for the device to which Configuration Manager installs the client.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeviceName<String>

Specifies the name of the device to which Configuration Manager installs the client.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ForceReinstall<Boolean>

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-IncludeDomainController<Boolean>

Indicates whether Configuration Manager authenticates and authorizes the client by using the clients domain controller.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a Configuration Manager client object. You can get a Configuration Manager client object by using the **Get-Client** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies the name of a Configuration Manager client.

Aliases	none
Required?	true
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteCode<String>

Specifies the site code for the Configuration Manager site that hosts this site system role.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Install a client

This command installs the client named RemoteClient05 on the Configuration Manager site that has the site code CM1.

```
PS C:\> Install-CMClient -Name "RemoteClient05" -SiteCode "CM1" -AlwaysInstallClient $True -
IncludeDomainController $True
```

Related topics

[Get-CMClientSetting](#)

[New-CMClientSetting](#)

[Remove-CMClientSetting](#)

[Set-CMClientSetting](#)

[Set-CMClientStatusSetting](#)

[Update-CMClientStatus](#)

[Get-CMDevice](#)

Invoke-CMAmtProvisioningDiscovery

Invoke-CMAmtProvisioningDiscovery

Checks whether computers have Intel AMT hardware.

Syntax

Parameter Set: SearchByDeviceNameMandatory

```
Invoke-CMAmtProvisioningDiscovery -DeviceName <String> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Invoke-CMAmtProvisioningDiscovery -DeviceId <String> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByDeviceValueMandatory

```
Invoke-CMAmtProvisioningDiscovery -Device <IResultObject> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByNameMandatory

```
Invoke-CMAmtProvisioningDiscovery -DeviceCollectionId <String> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Invoke-CMAmtProvisioningDiscovery -DeviceCollectionName <String> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Invoke-CMAmtProvisioningDiscovery -DeviceCollection <IResultObject> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Detailed Description

The **Invoke-CMAmtProvisioningDiscovery** cmdlet checks whether computers have Intel Active Management Technology (Intel AMT) hardware that supports out-of-band management. You can check individual computers or computers that belong to a Microsoft System Center 2012 Configuration Manager collection.

Parameters

-Device<IResultObject>

Specifies a device object in Configuration Manager. To obtain a device object, use the **Get-CMDevice** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeviceCollection<IResultObject>

Specifies a device collection object in Configuration Manager. To obtain a device collection object, use the **Get-CMDeviceCollection** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeviceCollectionId<String>

Specifies the ID of a device collection.

Aliases	none
Required?	true
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeviceCollectionName<String>

Specifies the name of a device collection.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeviceId<String>

Specifies the ID of a device.

Aliases	ResourceID
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeviceName<String>

Specifies the name of a device.

Aliases	Name
Required?	true
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Check a computer for Intel AMT hardware by using an ID

This command checks for Intel AMT-based hardware on a device that has the ID 16777230.

```
PS C:\> Invoke-CAmtProvisioningDiscovery -DeviceID "16777230"
```

Example 2: Check computers for Intel AMT hardware in a named device collection

This command checks for Intel AMT-based hardware on the devices that belong to the collection named Floor03.

```
PS C:\> Invoke-CMAmtProvisioningDiscovery -DeviceCollectionName "Floor03"
```

Example 3: Check for a computer for Intel AMT hardware by using a variable

The first command gets a device object by using the **Get-CMDevice** command, and stores it in the \$CMD variable.

The second command checks for Intel AMT-based technology on the device stored in the \$CMD variable.

```
PS C:\> $CMD = Get-CMDevice -Name "Accn023.Contoso.com"  
PS C:\> Invoke-CMAmtProvisioningDiscovery -Device $CMD
```

Related topics

[Update-CMAMTProvisioning](#)

[Remove-CMAmtProvisioningData](#)

[Get-CMDevice](#)

[Get-CMDeviceCollection](#)

Invoke-CMBaselineSummarization

Invoke-CMBaselineSummarization

Updates configuration baseline data.

Syntax

```
Invoke-CMBaselineSummarization [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Invoke-CMBaselineSummarization** cmdlet updates data in configuration baselines in Microsoft System Center 2012 Configuration Manager with the latest data from the site database. This action might take several minutes to complete.

You can use the **Set-CMBaselineSummarizationSchedule** cmdlet to configure a schedule by which the data is updated with the latest information from the site database.

Parameters

-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

Example 1: Update configuration baseline data

This command updates data in configuration baselines with the latest data from the site database.

```
PS C:\> Invoke-CMBaselineSummarization
```

Related topics

[Set-CMBaselineSummarizationSchedule](#)

[Get-CMBaselineSummarizationSchedule](#)

Invoke-CMClientNotification

Invoke-CMClientNotification

Sends a notification to client computers to trigger an immediate client action.

Syntax

Parameter Set: SearchByDeviceIdMandatory

```
Invoke-CMClientNotification -DeviceId <String> [-NotificationType {RequestMachinePolicyNow | RequestUsersPolicyNow} ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByDeviceNameMandatory

```
Invoke-CMClientNotification -DeviceName <String> [-NotificationType {RequestMachinePolicyNow | RequestUsersPolicyNow} ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByDeviceValueMandatory

```
Invoke-CMClientNotification -Device <IResultObject> [-NotificationType {RequestMachinePolicyNow | RequestUsersPolicyNow} ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Invoke-CMClientNotification -DeviceCollectionId <String> [-NotificationType {RequestMachinePolicyNow | RequestUsersPolicyNow} ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByNameMandatory

```
Invoke-CMClientNotification -DeviceCollectionName <String> [-NotificationType {RequestMachinePolicyNow | RequestUsersPolicyNow} ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Invoke-CMClientNotification -DeviceCollection <IResultObject> [-NotificationType {RequestMachinePolicyNow | RequestUsersPolicyNow} ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Invoke-CMClientNotification** cmdlet sends a notification to client computers to trigger an immediate client action. You can specify one or more client computers, or send a notification to all the computers in a specified device collection.

Parameters

-Device<IResultObject>

Specifies a **CMDevice** object. To obtain a **CMDevice** object, use the **Get-CMDevice** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeviceCollection<IResultObject>

Specifies a **CMDeviceCollection** object. To obtain a **CMDeviceCollection** object, use the **Get-CMDeviceCollection** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeviceCollectionId<String>

Specifies the ID of a device collection.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-DeviceCollectionName<String>

Specifies the name of a device collection.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeviceId<String>

Specifies the ID of a device.

Aliases	ResourceID
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeviceName<String>

Specifies a device name. You can specify a NetBIOS name or a fully qualified domain name (FQDN).

Aliases	Name
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-NotificationType<ClientNotificationType>

Specifies the type of notification to send. Valid values are:

-- RequestMachinePolicyNow. The client computer requests the latest machine policy from the management point. Machine policy includes configuration settings for a computer, or software updates that are deployed to a computer.

-- RequestUsersPolicyNow. The client computer requests the latest user policy from the management point. User policy includes applications or packages deployed for a user.

The acceptable values for this parameter are:

RequestMachinePolicyNow	
RequestUsersPolicyNow	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example: Send a notification to trigger an event

This command sends a notification of the type RequestMachinePolicyNow to the specified device.

```
PS C:\> Invoke-CMClientNotification -DeviceName "Computer073" -NotificationType  
RequestMachinePolicyNow
```

Related topics

[Get-CMDevice](#)

[Get-CMDeviceCollection](#)

Invoke-CMClientOperationSummarization

Invoke-CMClientOperationSummarization

Performs a Configuration Manager client operations summarization.

Syntax

```
Invoke-CMClientOperationSummarization [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Invoke-CMClientOperationSummarization** cmdlet performs a Microsoft System Center 2012 Configuration Manager client operations summarization immediately, instead of waiting for the next scheduled summarization. This cmdlet does not change the regular schedule for summarizations.

Parameters

-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

Example 1: Invoke summarization

This command performs a client operations summarization immediately.

```
PS C:\> Invoke-CMClientOperationSummarization
```

Related topics

[Clear-CMClientOperation](#)

Invoke-CMContentValidation

Invoke-CMContentValidation

Validates packages on a distribution point.

Syntax

Parameter Set: SearchByIdMandatory_Application

```
Invoke-CMContentValidation -ApplicationId <String[]> [-CollectionName <String> ] [-  
DisableDetectAssociatedContentDependencies] [-DistributionPointGroupName <String> ] [-  
DistributionPointName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory_BootImage

```
Invoke-CMContentValidation -BootImageId <String[]> [-CollectionName <String> ] [-  
DistributionPointGroupName <String> ] [-DistributionPointName <String> ] [-Confirm] [-  
WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory_DeploymentPackage

```
Invoke-CMContentValidation -DeploymentPackageId <String[]> [-CollectionName <String> ] [-  
DistributionPointGroupName <String> ] [-DistributionPointName <String> ] [-Confirm] [-  
WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory_DriverPackage

```
Invoke-CMContentValidation -DriverPackageId <String[]> [-CollectionName <String> ] [-  
DistributionPointGroupName <String> ] [-DistributionPointName <String> ] [-Confirm] [-  
WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory_OperatingSystemImage

```
Invoke-CMContentValidation -OperatingSystemImageId <String[]> [-CollectionName <String> ] [-  
DistributionPointGroupName <String> ] [-DistributionPointName <String> ] [-Confirm] [-  
WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory_OperatingSystemInstaller

```
Invoke-CMContentValidation -OperatingSystemInstallerId <String[]> [-CollectionName <String>  
] [-DistributionPointGroupName <String> ] [-DistributionPointName <String> ] [-Confirm] [-  
WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory_Package

```
Invoke-CMContentValidation -PackageId <String[]> [-CollectionName <String> ] [-  
DistributionPointGroupName <String> ] [-DistributionPointName <String> ] [-Confirm] [-  
WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory_TaskSequence

```
Invoke-CMContentValidation -TaskSequenceId <String[]> [-CollectionName <String> ] [-  
DistributionPointGroupName <String> ] [-DistributionPointName <String> ] [-Confirm] [-  
WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByNameMandatory_Application

Invoke-CMContentValidation -ApplicationName <String[]> [-CollectionName <String>] [-DisableDetectAssociatedContentDependencies] [-DistributionPointGroupName <String>] [-DistributionPointName <String>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SearchByNameMandatory_BootImage

Invoke-CMContentValidation -BootImageName <String[]> [-CollectionName <String>] [-DistributionPointGroupName <String>] [-DistributionPointName <String>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SearchByNameMandatory_DeploymentPackage

Invoke-CMContentValidation -DeploymentPackageName <String[]> [-CollectionName <String>] [-DistributionPointGroupName <String>] [-DistributionPointName <String>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SearchByNameMandatory_DriverPackage

Invoke-CMContentValidation -DriverPackageName <String[]> [-CollectionName <String>] [-DistributionPointGroupName <String>] [-DistributionPointName <String>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SearchByNameMandatory_OperatingSystemImage

Invoke-CMContentValidation -OperatingSystemImageName <String[]> [-CollectionName <String>] [-DistributionPointGroupName <String>] [-DistributionPointName <String>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SearchByNameMandatory_OperatingSystemInstaller

Invoke-CMContentValidation -OperatingSystemInstallerName <String[]> [-CollectionName <String>] [-DistributionPointGroupName <String>] [-DistributionPointName <String>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SearchByNameMandatory_Package

Invoke-CMContentValidation -PackageName <String[]> [-CollectionName <String>] [-DistributionPointGroupName <String>] [-DistributionPointName <String>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SearchByNameMandatory_TaskSequence

Invoke-CMContentValidation -TaskSequenceName <String[]> [-CollectionName <String>] [-DistributionPointGroupName <String>] [-DistributionPointName <String>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SearchByValueMandatory_Application

Invoke-CMContentValidation -Application <IResultObject> [-CollectionName <String>] [-DisableDetectAssociatedContentDependencies] [-DistributionPointGroupName <String>] [-DistributionPointName <String>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SearchByValueMandatory_BootImage

Invoke-CMContentValidation -BootImage <IResultObject> [-CollectionName <String>] [-DistributionPointGroupName <String>] [-DistributionPointName <String>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SearchByValueMandatory_DeploymentPackage

Invoke-CMContentValidation -DeploymentPackage <IResultObject> [-CollectionName <String>] [-

```
DistributionPointGroupName <String> ] [-DistributionPointName <String> ] [-Confirm] [-
WhatIf] [ <CommonParameters>]
```

```
Parameter Set: SearchByValueMandatory_DriverPackage
```

```
Invoke-CMContentValidation -DriverPackage <IResultObject> [-CollectionName <String> ] [-
DistributionPointGroupName <String> ] [-DistributionPointName <String> ] [-Confirm] [-
WhatIf] [ <CommonParameters>]
```

```
Parameter Set: SearchByValueMandatory_OperatingSystemImage
```

```
Invoke-CMContentValidation -OperatingSystemImage <IResultObject> [-CollectionName <String> ]
[-DistributionPointGroupName <String> ] [-DistributionPointName <String> ] [-Confirm] [-
WhatIf] [ <CommonParameters>]
```

```
Parameter Set: SearchByValueMandatory_OperatingSystemInstaller
```

```
Invoke-CMContentValidation -OperatingSystemInstaller <IResultObject> [-CollectionName
<String> ] [-DistributionPointGroupName <String> ] [-DistributionPointName <String> ] [-
Confirm] [-WhatIf] [ <CommonParameters>]
```

```
Parameter Set: SearchByValueMandatory_Package
```

```
Invoke-CMContentValidation -Package <IResultObject> [-CollectionName <String> ] [-
DistributionPointGroupName <String> ] [-DistributionPointName <String> ] [-Confirm] [-
WhatIf] [ <CommonParameters>]
```

```
Parameter Set: SearchByValueMandatory_TaskSequence
```

```
Invoke-CMContentValidation -TaskSequence <IResultObject> [-CollectionName <String> ] [-
DistributionPointGroupName <String> ] [-DistributionPointName <String> ] [-Confirm] [-
WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Invoke-CMContentValidation** cmdlet validates one or more packages on a distribution point. Validating the content ensures that the entire set of files transferred successfully to the distribution point.

Parameters

-Application<IResultObject>

Specifies a Configuration Manager application object. To obtain a **CMApplication** object, use the **Get-CMApplication** cmdlet.

Aliases	none
Required?	true
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ApplicationId<String[]>

Specifies an array of application IDs. These IDs are GUIDs as strings.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ApplicationName<String[]>

Specifies an array of application names.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-BootImage<IResultObject>

Specifies a boot image object. To obtain a **CMBootImage** object, use the **Get-CMBootImage** cmdlet.

Aliases	none
Required?	true
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-BootImageId<String[]>

Specifies an array of IDs of boot images.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-BootImageName<String[]>

Specifies an array of names of boot images.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CollectionName<String>

Specifies the name of a Configuration Manager collection.

Aliases	none
Required?	false
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeploymentPackage<IResultObject>

Specifies a deployment package object. To obtain a **CMDeploymentPackage** object, use the **Get-
CMDeploymentPackage** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeploymentPackageld<String[]>

Specifies an array of IDs of deployment packages.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeploymentPackageName<String[]>

Specifies an array of names of deployment packages.

Aliases	none
Required?	true

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DisableDetectAssociatedContentDependencies

Indicates that Configuration Manager automatically detects associated content dependencies and adds the associated content to the distribution for applications.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DistributionPointGroupName<String>

Specifies the name of a distribution point group.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DistributionPointName<String>

Specifies the name of a distribution point that is associated with the content.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DriverPackage<IResultObject>

Specifies a driver package object. To obtain a **CMDriverPackage** object, use the **Get-CMDriverPackage** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DriverPackageld<String[]>

Specifies an array of IDs of driver packages.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DriverPackageName<String[]>

Specifies an array of names of driver packages.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OperatingSystemImage<IResultObject>

Specifies an operating system image object. To obtain a **CMOperatingSystemImage** object, use the **Get-CMOperatingSystemImage** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OperatingSystemImageId<String[]>

Specifies an array of IDs of operating system images.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OperatingSystemImageName<String[]>

Specifies an array of names of operating system images.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OperatingSystemInstaller<IResultObject>

Specifies an operating system installer object. To obtain a **CMOperatingSystemInstaller** object, use the **Get-CMOperatingSystemInstaller** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OperatingSystemInstallerId<String[]>

Specifies an array of IDs of operating system installers.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OperatingSystemInstallerName<String[]>

Specifies an array of names of operating system installers.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Package<IResultObject>

Specifies a package object. To obtain a **CMPackage** object, use the **Get-CMPackage** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Packageld<String[]>

Specifies an array of IDs of packages.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PackageName<String[]>

Specifies an array of names of packages.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-TaskSequence<IResultObject>

Specifies a task sequence object. To obtain a **CMTaskSequence** object, use the **Get-CMTaskSequence** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-TaskSequenceId<String[]>

Specifies an array of IDs of task sequences.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-TaskSequenceName<String[]>

Specifies an array of names of task sequences.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Validate content for an application

This command validates the package for the application named Dict.app on the distribution point named DPServer01.

```
PS C:\> Invoke-CMContentValidation -ApplicationName "Dict.app" -DistributionPointName "DPServer01"
```

Related topics

[Get-CMApplication](#)

[Get-CMDeploymentPackage](#)

[Get-CMDriverPackage](#)

[Get-CMOperatingSystemImage](#)

[Get-CMOperatingSystemInstaller](#)

[Get-CMTaskSequence](#)

[Start-CMContentDistribution](#)

Invoke-CMDeploymentSummarization

Invoke-CMDeploymentSummarization

Runs a Configuration Manager deployment summarization.

Syntax

Parameter Set: SearchByIdMandatory

```
Invoke-CMDeploymentSummarization -DeploymentId <String> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByNameMandatory

```
Invoke-CMDeploymentSummarization -CollectionName <String> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Invoke-CMDeploymentSummarization -InputObject <IResultObject> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Detailed Description

The **Invoke-CMDeploymentSummarization** cmdlet runs a Microsoft System Center 2012 Configuration Manager deployment summarization as soon as possible. Summarization compiles information about current deployment of software from the System Center 2012 Configuration Manager site database. By default, System Center 2012 Configuration Manager runs this summarization every four hours. If you use this cmdlet to create the summarization immediately, it does not interfere with the regular schedule of creating the current summarization.

You can specify a deployment summarization by ID or by collection or you can specify a deployment summarization object.

Parameters

-CollectionName<String>

Specifies a name of a collection.

Aliases	none
Required?	true

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeploymentId<String>

Specifies an ID of a deployment.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a deployment summarization object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Invoke a deployment summarization

This command runs a deployment summarization for the collection named CMWest.

```
PS C:\> Invoke-CMDeploymentSummarization -CollectionName "CMWest"
```

Invoke-CMDeviceCollectionUpdate

Invoke-CMDeviceCollectionUpdate

Starts an update operation for a device collection.

Syntax

Parameter Set: SearchByIdMandatory

```
Invoke-CMDeviceCollectionUpdate -CollectionId <String> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByNameMandatory

```
Invoke-CMDeviceCollectionUpdate -Name <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Invoke-CMDeviceCollectionUpdate -InputObject <IResultObject> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Detailed Description

The **Invoke-CMDeviceCollectionUpdate** cmdlet starts an operation to update the membership of a device collection. This operation discovers new devices in the site hierarchy and adds them to the collection.

Parameters

-CollectionId<String>

Specifies the device collection to update by using an ID.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies the device collection to update by using a device collection object. To obtain a device collection object, use the [Get-CMDeviceCollection](#) cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies the device collection to update by using a name.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Invoke a device collection update by using a name

This command starts a device collection update on the device collection named All Systems.

```
PS C:\> Invoke-CMDeviceCollectionUpdate -Name "All Systems"
```

Related topics

[Export-CMDeviceCollection](#)

[Get-CMDeviceCollection](#)

[Import-CMDeviceCollection](#)

[New-CMDeviceCollection](#)

[Set-CMDeviceCollection](#)

Invoke-CMDeviceRetire

Invoke-CMDeviceRetire

Retires devices.

Syntax

Parameter Set: SearchByNameMandatory

```
Invoke-CMDeviceRetire -DeviceName <String[]> [-Force] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Invoke-CMDeviceRetire -DeviceId <String[]> [-Force] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Invoke-CMDeviceRetire -InputObject <IResultObject> [-Force] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Detailed Description

The **Invoke-CMDeviceRetire** cmdlet retires one or more devices. A retired device is no longer active in Microsoft System Center 2012 Configuration Manager. It does not receive new policies or policy updates. Retired devices remain listed until a maintenance task removes them.

Parameters

-DeviceId<String[]>

Specifies an array of device IDs.

Aliases	ResourceId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeviceName<String[]>

Specifies an array of device names.

Aliases	Name
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a **CMDevice** object. To obtain a **CMDevice** object, use the **Get-CMDevice** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Retire a named device

This command retires the computer named Computer073.

```
PS C:\> Invoke-CMDeviceRetire -DeviceName "Computer073"
```

Related topics

[Get-CMDevice](#)



Invoke-CMDeviceWipe

Invoke-CMDeviceWipe

Performs a wipe of a modern device.

Syntax

Parameter Set: SearchByIdMandatory

```
Invoke-CMDeviceWipe -DeviceId <String[]> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByNameMandatory

```
Invoke-CMDeviceWipe -DeviceName <String[]> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Invoke-CMDeviceWipe -InputObject <IResultObject> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Invoke-CMDeviceWipe** cmdlet performs a complete wipe of all applications and data on a modern device. This cmdlet returns one or more devices to the factory configuration.

Parameters

-DeviceId<String[]>

Specifies an array of device IDs.

Aliases	ResourceId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeviceName<String[]>

Specifies an array of device names.

Aliases	Name
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a **CMDevice** object. To obtain a **CMDevice** object, use the **Get-CMDevice** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Return a device to its factory configuration

This command returns the named device to its factory configuration.

```
PS C:\> Invoke-CMDeviceWipe -DeviceName "WindowsPhone_17"
```

Related topics

[Set-CMDeviceOwnership](#)

[Get-CMDevice](#)

Invoke-CMEndpointProtectionScan

Invoke-CMEndpointProtectionScan

Invokes a scan to detect malware on one or more devices in the Configuration Manager hierarchy.

Syntax

Parameter Set: SearchByDeviceIdMandatory

```
Invoke-CMEndpointProtectionScan -DeviceId <String> [-ScanType {Full | Quick} ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByDeviceNameMandatory

```
Invoke-CMEndpointProtectionScan -DeviceName <String> [-ScanType {Full | Quick} ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByDeviceValueMandatory

```
Invoke-CMEndpointProtectionScan -Device <IResultObject> [-ScanType {Full | Quick} ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Invoke-CMEndpointProtectionScan -DeviceCollectionId <String> [-ScanType {Full | Quick} ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByNameMandatory

```
Invoke-CMEndpointProtectionScan -DeviceCollectionName <String> [-ScanType {Full | Quick} ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Invoke-CMEndpointProtectionScan -DeviceCollection <IResultObject> [-ScanType {Full | Quick} ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Invoke-CMEndpointProtectionScan** cmdlet invokes a System Center 2012 Endpoint Protection scan that is outside of any scheduled scans. You can specify the device or collection by using its name, ID, or by specifying an object that represents the device or collection.

For more information about how Configuration Manager supports Endpoint Protection, see [Endpoint Protection in Configuration Manager](http://go.microsoft.com/fwlink/?LinkId=268427) (http://go.microsoft.com/fwlink/?LinkId=268427) on TechNet.

Parameters

-Device<IResultObject>

Specifies the device that is scanned for malware.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeviceCollection<IResultObject>

Specifies an object that represents a device collection whose members are scanned for malware.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeviceCollectionId<String>

Specifies the ID of a device collection whose members are scanned for malware.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeviceCollectionName<String>

Specifies the name of a device collection whose members are scanned for malware.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeviceId<String>

Specifies the ID of a device that is scanned for malware.

Aliases	ResourceID
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeviceName<String>

Specifies the name of a device that is scanned for malware.

Aliases	Name
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ScanType<ScanType>

Specifies a full or a quick scan. A full scan looks at every location on the device. A quick scan looks at only those locations where malware is most likely to appear.

The acceptable values for this parameter are:

Full	
Quick	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Invoke a full Endpoint Protection scan

This command invokes a full Endpoint Protection scan of the device named CMCEN-DIST02.

```
PS C:\> Invoke-CMEndpointProtectionScan -DeviceName "CMCEN-DIST02" -ScanType Full
```

Related topics

[Get-CMEndpointProtectionPoint](#)

Invoke-CMEndpointProtectionSummarization

Invoke-CMEndpointProtectionSummarization

Retrieves summary status data about Endpoint Protection.

Syntax

```
Invoke-CMEndpointProtectionSummarization [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Invoke-CMEndpointProtectionSummarization** cmdlet retrieves summary status data about System Center 2012 Endpoint Protection in Microsoft System Center 2012 Configuration Manager. This data helps you monitor Endpoint Protection in your System Center 2012 Configuration Manager hierarchy.

For more information about configuring and monitoring Endpoint Protection, see [How To Monitor Endpoint Configuration In Configuration Manager](http://go.microsoft.com/fwlink/?linkid=268428) (http://go.microsoft.com/fwlink/?linkid=268428) on TechNet.

Parameters

-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

Example 1: Invoke summarization for Endpoint Protection

This command gets summary data about Endpoint Protection status after you confirm that you want to run the command.

```
PS C:\> Invoke-CMEndpointProtectionSummarization -Confirm
```

Related topics

[Get-CMEndpointProtectionSummarizationSchedule](#)

[Set-CMEndpointProtectionSummarizationSchedule](#)

Invoke-CMForestDiscovery

Invoke-CMForestDiscovery

Starts a forest discovery operation in Active Directory.

Syntax

Parameter Set: SearchBySiteCodeMandatory

```
Invoke-CMForestDiscovery -SiteCode <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Invoke-CMForestDiscovery** cmdlet starts a forest discovery operation in Active Directory. A forest discovery returns a list of Active Directory sites, subnets, and domains.

Parameters

-SiteCode<String>

Specifies the site that is the starting point for the forest discovery operation.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Invoke a forest discovery operation

This command invokes a forest discovery operation in Active Directory starting at the site that has the site code CM4.

```
PS C:\> Invoke-CMForestDiscovery -SiteCode "CM4"
```

Invoke-CMRemoteControl

Invoke-CMRemoteControl

Enables remote control on computers.

Syntax

Parameter Set: InvokeDeviceByIdMandatory

```
Invoke-CMRemoteControl -DeviceId <String[]> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: InvokeDeviceByNameMandatory

```
Invoke-CMRemoteControl -DeviceName <String[]> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: InvokeDeviceByValueMandatory

```
Invoke-CMRemoteControl -Device <IResultObject> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: InvokeSiteStatusByNameMandatory

```
Invoke-CMRemoteControl -SiteSystemServerName <String[]> [-SiteCode <String[]> ] [-SiteSystemRole <String[]> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: InvokeSiteStatusByValueMandatory

```
Invoke-CMRemoteControl -SiteStatus <IResultObject> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Invoke-CMRemoteControl** cmdlet enables remote control on the computers that you want to use in a remote control session. You can enable remote control on computers by specifying the ID or name of the computers, the site systems, or the site status.

Use remote control in Microsoft System Center 2012 Configuration Manager to remotely administer, provide assistance, or view any client computer in the hierarchy. You can use remote control to troubleshoot hardware and software configuration problems on client computers and to provide help desk support when access to the computer of a user is required. System Center 2012 Configuration Manager supports the remote control of workgroup computers and computers that are joined to an Active Directory domain.

Parameters

-Device<IResultObject>

Specifies a **CMDevice** object. To obtain a **CMDevice** object, use the **Get-CMDevice** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeviceId<String[]>

Specifies an array of device IDs.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeviceName<String[]>

Specifies an array of device names.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteCode<String[]>

Specifies an array of site codes of Configuration Manager sites that host the site system roles.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteStatus<IResultObject>

Specifies a **CMSiteStatusMessage** object. To obtain a **CMSiteStatusMessage** object, use the **Get-CMSiteStatusMessage** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteSystemRole<String[]>

Specifies an array of Configuration Manager roles that the site system performs.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteSystemServerName<String[]>

Specifies an array of fully qualified domain names (FQDN) of the servers that host the site system roles.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Enable remote control on a computer

This command enables remote control on the computer named CMCEN-DIST02.

```
PS C:\> Invoke-CMRemoteControl -DeviceName "CMCEN-DIST02"
```

Related topics

[Get-CMDevice](#)

[Get-CMSiteStatusMessage](#)

Invoke-CMReport

Invoke-CMReport

Invokes a report about data and operations in Configuration Manager.

Syntax

Parameter Set: Default

```
Invoke-CMReport -ReportPath <String> -SiteCode <String> [-OutputFormat <String> ] [-  
ReportParameter <Hashtable> ] [-SrsServerName <String> ] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Detailed Description

The **Invoke-CMReport** cmdlet invokes a Microsoft SQL Reporting Services report that displays information about data and operations in Microsoft System Center 2012 Configuration Manager. The reporting services point is a site system role that you install on a server that is running Microsoft SQL Reporting Services.

Parameters

-OutputFormat<String>

Specifies an output format in Configuration Manager.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ReportParameter<Hashtable>

Specifies report parameters as key-value pairs.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ReportPath<String>

Specifies the path to a folder where reports are stored.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteCode<String>

Specifies a site code in Configuration Manager.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SrsServerName<String>

Specifies a name of a SQL Reporting Services server.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Invoke a report

This command invokes a report by using a report path and a site code.

```
PS C:\> Invoke-CMReport -ReportPath "/Reports/Data" -SiteCode "CM4"
```

Invoke-CMSecondarySiteUpgrade

Invoke-CMSecondarySiteUpgrade

Invokes a secondary site upgrade.

Syntax

Parameter Set: SearchBySiteCodeMandatory

```
Invoke-CMSecondarySiteUpgrade -SiteCode <String> [-Force] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByNameMandatory

```
Invoke-CMSecondarySiteUpgrade -Name <String> [-Force] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Invoke-CMSecondarySiteUpgrade -InputObject <IResultObject> [-Force] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Detailed Description

The **Invoke-CMSecondarySiteUpgrade** cmdlet invokes a secondary site upgrade that is outside of any scheduled upgrades. You can specify the site upgrade by using its name or Id, or by specifying an object that represents the site upgrade.

Parameters

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a secondary site upgrade object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies the name of a secondary site.

Aliases	SiteName
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteCode<String>

Specifies a site code for the upgrade site.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Invoke a secondary site upgrade by using a site name

This example invokes a secondary site upgrade from a site named ClientSecSiteUpgrade03. Because the *Force* parameter is specified, Configuration Manager invokes the update automatically and does not prompt you for confirmation.

```
PS C:\> Invoke-CMSecondarySiteUpgrade -SiteName "ClientSecSiteUpgrade03" -Force
```

Related topics

[New-CMSecondarySite](#)

[Remove-CMSecondarySite](#)

Invoke-CMSoftwareUpdateAutoDeploymentRule

Invoke-CMSoftwareUpdateAutoDeploymentRule

Runs a Configuration Manager deployment rule for automatic software updates.

Syntax

Parameter Set: SearchByIdMandatory

```
Invoke-CMSoftwareUpdateAutoDeploymentRule -Id <String[]> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByNameMandatory

```
Invoke-CMSoftwareUpdateAutoDeploymentRule -Name <String> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Invoke-CMSoftwareUpdateAutoDeploymentRule -InputObject <IResultObject> [-Confirm] [-WhatIf]  
[ <CommonParameters>]
```

Detailed Description

The **Invoke-CMSoftwareUpdateAutoDeploymentRule** cmdlet runs a Microsoft System Center 2012 Configuration Manager deployment rule for automatic software updates immediately instead of according to its schedule.

System Center 2012 Configuration Manager uses rules to manage automatic deployment of software updates. When a rule runs, System Center 2012 Configuration Manager adds updates that qualify for the rule to a software update group. The System Center 2012 Configuration Manager server downloads content files and copies them to distribution points, and then updates client computers.

You can specify rules to run by ID or by name, or specify a rule object by using the **Get-CMSoftwareUpdateAutoDeploymentRule** cmdlet. You cannot run a disabled rule. You can use the **Enable-CMSoftwareUpdateAutoDeploymentRule** cmdlet to enable a rule and then run it.

Parameters

-Id<String[]>

Specifies an array of IDs for rules for automatic deployment of software updates. This value is the **AutoDeploymentID** property of the deployment rule object.

Aliases	AutoDeploymentId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a software update automatic deployment rule object. To obtain a deployment rule object, use the **Get- CMSoftwareUpdateAutoDeploymentRule** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies a name of a rule for automatic deployment of software updates.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Invoke a deployment rule

This command runs a rule called Weekly Security Updates.

```
PS C:\> Invoke-CMSoftwareUpdateAutoDeploymentRule -Name "Weekly Security Updates"
```

Related topics

[Disable-CMSoftwareUpdateAutoDeploymentRule](#)

[Enable-CMSoftwareUpdateAutoDeploymentRule](#)

[Get-CMSoftwareUpdateAutoDeploymentRule](#)

[Remove-CMSoftwareUpdateAutoDeploymentRule](#)

Invoke-CMSoftwareUpdateSummarization

Invoke-CMSoftwareUpdateSummarization

Runs the Configuration Manager software update summarization.

Syntax

```
Invoke-CMSoftwareUpdateSummarization [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Invoke-CMSoftwareUpdateSummarization** cmdlet runs the software update summarization in Microsoft System Center 2012 Configuration Manager immediately. System Center 2012 Configuration Manager summarizes software update status on a regular schedule. This cmdlet does not reset the time for the next automatic summarization.

You can use the **Get-CMSoftwareUpdateSummarizationSchedule** cmdlet to view the current schedule and the **Set-CMSoftwareUpdateSummarizationSchedule** cmdlet to change the schedule.

Parameters

-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

Example 1: Run software update summarization

This command runs the software update summarization immediately, instead of waiting for the next scheduled time.

```
PS C:\> Invoke-CMSoftwareUpdateSummarization
```

Related topics

[Get-CMSoftwareUpdateSummarizationSchedule](#)

[Set-CMSoftwareUpdateSummarizationSchedule](#)

Invoke-CMUserCollectionUpdate

Invoke-CMUserCollectionUpdate

Invokes an update to a user collection.

Syntax

Parameter Set: SearchByIdMandatory

```
Invoke-CMUserCollectionUpdate -CollectionId <String> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByNameMandatory

```
Invoke-CMUserCollectionUpdate -Name <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Invoke-CMUserCollectionUpdate -InputObject <IResultObject> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Detailed Description

The **Invoke-CMUserCollectionUpdate** cmdlet invokes an update to a user collection.

User collections represent logical groupings of users in Microsoft System Center 2012 Configuration Manager, such as the default All Users collection and All User Groups collection.

Parameters

-CollectionId<String>

Specifies a collection ID in Configuration Manager.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a **CMUserCollectionUpdate** object. To obtain a **CMUserCollectionUpdate** object, use the **Get-CMUserCollection** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies a name of a user collection.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Invoke a user collection update

This command invokes an update of the All Local Users collection.

```
PS C:\> Invoke-CMUserCollectionUpdate -Name "All Local Users"
```

Related topics

[Get-CMUserCollection](#)

Lock-CMObject

Lock-CMObject

Locks global objects in Configuration Manager.

Syntax

Parameter Set: ByValue

```
Lock-CMObject [-InputObject] <IResultObject[]> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Lock-CMObject** cmdlet acquires locks of one or more objects in Microsoft System Center 2012 Configuration Manager. You can use the *InputObject* parameter to specify the input to this cmdlet, or you can pipe the input to this cmdlet.

Serialized Editing of Distributed Objects (SEDO) in System Center 2012 Configuration Manager provide a mechanism for assigning and unassigning locks to global System Center 2012 Configuration Manager objects in the context of a site, computer and user. When you use the Administrator Console, SEDO lock and unlock functions occur automatically. If you use Windows PowerShell cmdlets in a multiple-site environment, we recommend that you lock and unlock objects to prevent inadvertent overwriting of data. If you want to edit and save a SEDO-enabled object, you must lock the object. When you obtain the lock, the lock is assigned to you, your computer and the site in which the computer resides. While the lock is assigned to you, no other user or computer can edit the object until you release the lock.

Parameters

-InputObject<IResultObject[]>

Specifies an array of Configuration Manager objects output from another cmdlet.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

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

Example 1: Lock a global object

The first command gets the driver package object that has the ID CM100042, and stores the object in the \$CIObj variable.

The second command locks the object stored in \$CIObj.

```
PS C:\> $CIObj = Get-CMDriverPackage -Id "CM100042"  
PS C:\> Lock-CMObject $CIObj
```

Related topics

[Unlock-CMObject](#)

Merge-CMAntimalwarePolicy

Merge-CMAntimalwarePolicy

Merges antimalware policies for Endpoint Protection.

Syntax

Parameter Set: MergePolicyById

```
Merge-CMAntimalwarePolicy -BasePolicyId <String> -NewPolicyName <String> -OtherPolicyId <String[]> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: MergePolicyByName

```
Merge-CMAntimalwarePolicy -BasePolicyName <String> -NewPolicyName <String> -OtherPolicyName <String[]> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Merge-CMAntimalwarePolicy** cmdlet merges two or more existing antimalware policies for System Center 2012 Endpoint Protection. This cmdlet can help you consolidate antimalware policies to reduce the number of policies that you manage.

When you merge antimalware policies, the merge operation considers the priority you have configured for each antimalware policy. If two settings conflict, the highest-priority option takes precedence. A merge operation also merges some settings, such as exclusion lists from different antimalware policies.

Parameters

-BasePolicyId<String>

Specifies the antimalware policy that you are merging with other policies by using an ID.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-BasePolicyName<String>

Specifies the antimalware policy that you are merging with other policies by using a name.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-NewPolicyName<String>

Specifies the name of the new antimalware policy that contains the merged policies.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OtherPolicyId<String[]>

Specifies an array of policies to merge with the base policy by using the policy IDs.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OtherPolicyName<String[]>

Specifies an array of policies to merge with the base policy by using the policy names.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Merge antimalware policies

This command merges the antimalware policies in the array named ContosoPoliciesNewFor2013 with an existing antimalware policy named ContosoPolicy. The cmdlet gives the name ContosoPolicy2013 to the merged policy.

```
PS C:\> Merge-CMAntimalwarePolicy -BasePolicyName "ContosoPolicy" -NewPolicyName  
"ContosoPolicy2013" -OtherPolicyName "ContosoPoliciesNewFor2013"
```

Related topics

[Export-CMAntimalwarePolicy](#)

[Get-CMAntiMalwarePolicy](#)

[Import-CMAntimalwarePolicy](#)

[New-CMAntimalwarePolicy](#)

[Remove-CMAntiMalwarePolicy](#)

[Set-CMAntiMalwarePolicy](#)

Merge-CMConflictingRecord

Merge-CMConflictingRecord

Merges a new Configuration Manager client record with a conflicting client record.

Syntax

Parameter Set: SearchBySiteCode

```
Merge-CMConflictingRecord -SiteCode <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Merge-CMConflictingRecord -Id <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByNameMandatory

```
Merge-CMConflictingRecord -Name <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Merge-CMConflictingRecord** cmdlet merges a new Microsoft System Center 2012 Configuration Manager client record with a conflicting client record that has the same hardware information.

When System Center 2012 Configuration Manager recognizes a new client, it uses hardware information to check whether it previously created a record for that computer. For example, you might have reinstalled the operating system. The previous client record still exists with the same hardware information. If you manually resolve conflicts, you have the option to merge the new record with the existing record, create a new record, or create a record as a blocked record. You can also configure System Center 2012 Configuration Manager to resolve conflicts automatically. For more information, see [Configuring Settings for Client Management in Configuration Manager](http://go.microsoft.com/fwlink/?LinkId=269847) (<http://go.microsoft.com/fwlink/?LinkId=269847>) on TechNet.

You can specify conflicting records by using a name or ID or you can specify a site code to merge each of the unresolved conflicting records for that site.

Parameters

-Id<String>

Specifies an ID for the conflicting records.

Aliases	Smsid
---------	-------

Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies a name for the conflicting records.

Aliases	AgentName
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteCode<String>

Specifies a site code for a Configuration Manager site. This cmdlet merges the conflicting records for this site.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Merge conflicting records for a site

This command merges each of the conflicting records for the specified System Center 2012 Configuration Manager site.

```
PS C:\> Merge-CMConflictingRecords -SiteCode "CM2"
```

Example 2: Merge records for a named conflict

This command merges the conflicting records named CR07.

```
PS C:\> Merge-CMConflictingRecords -Name "CR07"
```

Related topics

[Block-CMConflictingRecord](#)

[Get-CMConflictingRecord](#)

Move-CMObject

Move-CMObject

Moves a Configuration Manager object into a different folder.

Syntax

Parameter Set: SearchByIdMandatory

```
Move-CMObject -FolderPath <String> -ObjectId <String[]> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByObjectMandatory

```
Move-CMObject -FolderPath <String> -InputObject <IResultObject[]> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Detailed Description

The **Move-CMObject** cmdlet moves a Microsoft System Center 2012 Configuration Manager object into a different folder. Specify the object to move and the destination folder. Because an object exists in only one folder, the cmdlet does not specify the current folder.

Parameters

-FolderPath<String>

Specifies a destination folder path, in the following format: *AAA:\<object type>\folder\subfolder\subfolder*, where AAA represents the Configuration Manager site code. For example, a folder called LOB Apps for an application node at a site designated CM1 has the following file path: CM1:\Application\LOB Apps.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject[]>

Specifies an array of Configuration Manager objects to move.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ObjectId<String[]>

Specifies an array of IDs of objects to move.

Aliases	InstanceKey
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Move an object

This command moves the object that has the specified ID to the folder GKP:\Application\TestFolder.

```
PS C:\> Move-CMObject -FolderPath "GKP:\Application\TestFolder" -ObjectId "209224563"
```

Related topics

[Lock-CMObject](#)

[Unlock-CMObject](#)

New-CMAccessAccount

New-CMAccessAccount

Adds users or groups to an access account.

Syntax

Parameter Set: SearchByApplicationName

```
New-CMAccessAccount -Access {Change | FullControl | NoAccess | Read} -AccountType  
{Administrator | Guest | User | WindowsGroup | WindowsUser} -ApplicationName <String> [-  
UserName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByApplication

```
New-CMAccessAccount -Access {Change | FullControl | NoAccess | Read} -AccountType  
{Administrator | Guest | User | WindowsGroup | WindowsUser} -Application <IResultObject> [-  
UserName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByApplicationId

```
New-CMAccessAccount -Access {Change | FullControl | NoAccess | Read} -AccountType  
{Administrator | Guest | User | WindowsGroup | WindowsUser} -ApplicationId <String> [-  
UserName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByBootImage

```
New-CMAccessAccount -Access {Change | FullControl | NoAccess | Read} -AccountType  
{Administrator | Guest | User | WindowsGroup | WindowsUser} -BootImage <IResultObject> [-  
UserName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByBootImageId

```
New-CMAccessAccount -Access {Change | FullControl | NoAccess | Read} -AccountType  
{Administrator | Guest | User | WindowsGroup | WindowsUser} -BootImageId <String> [-UserName  
<String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByBootImageName

```
New-CMAccessAccount -Access {Change | FullControl | NoAccess | Read} -AccountType  
{Administrator | Guest | User | WindowsGroup | WindowsUser} -BootImageName <String> [-  
UserName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByDriverPackage

```
New-CMAccessAccount -Access {Change | FullControl | NoAccess | Read} -AccountType  
{Administrator | Guest | User | WindowsGroup | WindowsUser} -DriverPackage <IResultObject>  
[-UserName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByDriverPackageId

```
New-CMAccessAccount -Access {Change | FullControl | NoAccess | Read} -AccountType  
{Administrator | Guest | User | WindowsGroup | WindowsUser} -DriverPackageId <String> [-  
UserName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByDriverPackageName

New-CMAccessAccount -Access {Change | FullControl | NoAccess | Read} -AccountType {Administrator | Guest | User | WindowsGroup | WindowsUser} -DriverPackageName <String> [-UserName <String>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SearchByOSImage

New-CMAccessAccount -Access {Change | FullControl | NoAccess | Read} -AccountType {Administrator | Guest | User | WindowsGroup | WindowsUser} -OperatingSystemImage <IResultObject> [-UserName <String>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SearchByOSImageId

New-CMAccessAccount -Access {Change | FullControl | NoAccess | Read} -AccountType {Administrator | Guest | User | WindowsGroup | WindowsUser} -OperatingSystemImageId <String> [-UserName <String>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SearchByOSImageName

New-CMAccessAccount -Access {Change | FullControl | NoAccess | Read} -AccountType {Administrator | Guest | User | WindowsGroup | WindowsUser} -OperatingSystemImageName <String> [-UserName <String>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SearchByOSInstaller

New-CMAccessAccount -Access {Change | FullControl | NoAccess | Read} -AccountType {Administrator | Guest | User | WindowsGroup | WindowsUser} -OperatingSystemInstaller <IResultObject> [-UserName <String>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SearchByOSInstallerId

New-CMAccessAccount -Access {Change | FullControl | NoAccess | Read} -AccountType {Administrator | Guest | User | WindowsGroup | WindowsUser} -OperatingSystemInstallerId <String> [-UserName <String>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SearchByOSInstallerName

New-CMAccessAccount -Access {Change | FullControl | NoAccess | Read} -AccountType {Administrator | Guest | User | WindowsGroup | WindowsUser} -OperatingSystemInstallerName <String> [-UserName <String>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SearchByPackage

New-CMAccessAccount -Access {Change | FullControl | NoAccess | Read} -AccountType {Administrator | Guest | User | WindowsGroup | WindowsUser} -Package <IResultObject> [-UserName <String>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SearchByPackageId

New-CMAccessAccount -Access {Change | FullControl | NoAccess | Read} -AccountType {Administrator | Guest | User | WindowsGroup | WindowsUser} -PackageId <String> [-UserName <String>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SearchByPackageName

New-CMAccessAccount -Access {Change | FullControl | NoAccess | Read} -AccountType {Administrator | Guest | User | WindowsGroup | WindowsUser} -PackageName <String> [-UserName <String>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SearchBySoftwareUpdateDeploymentPackage

New-CMAccessAccount -Access {Change | FullControl | NoAccess | Read} -AccountType

```
{Administrator | Guest | User | WindowsGroup | WindowsUser} -SoftwareUpdateDeploymentPackage
<IResultObject> [-UserName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

```
Parameter Set: SearchBySoftwareUpdateDeploymentPackageId
New-CMAccessAccount -Access {Change | FullControl | NoAccess | Read} -AccountType
{Administrator | Guest | User | WindowsGroup | WindowsUser} -
SoftwareUpdateDeploymentPackageId <String> [-UserName <String> ] [-Confirm] [-WhatIf] [
<CommonParameters>]
```

```
Parameter Set: SearchBySoftwareUpdateDeploymentPackageName
New-CMAccessAccount -Access {Change | FullControl | NoAccess | Read} -AccountType
{Administrator | Guest | User | WindowsGroup | WindowsUser} -
SoftwareUpdateDeploymentPackageName <String> [-UserName <String> ] [-Confirm] [-WhatIf] [
<CommonParameters>]
```

Detailed Description

The **New-CMAccessAccount** cmdlet adds users or groups to an access account.

An access account is a list of users or groups that can access an established service or application that is located on a distribution point. For example, members in the Software Update Point Connection access account can access two services to manage software updates: Windows Server Update Services (WSUS) and WSUS Synchronization Manager.

Parameters

-Access<AccessRight>

Specifies the access rights that are associated with an access account. Valid values are: No Access, Read, Change, and Full Control.

The acceptable values for this parameter are:

Change	
FullControl	
NoAccess	
Read	

Aliases	none
Required?	true

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AccountType<AccessAccountType>

Specifies an account type. Valid values are: Guest, User, WindowsGroup, and WindowsUser.

The acceptable values for this parameter are:

Administrator	
Guest	
User	
WindowsGroup	
WindowsUser	

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Application<IResultObject>

Specifies a deployed application object. You can get an application object by using the **Get-CAApplication** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ApplicationId<String>

Specifies the ID of an application.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ApplicationName<String>

Specifies the name of an application.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-BootImage<IResultObject>

Specifies a boot image object. A boot image object contains the Windows files that are required to prepare a computer for the installation of an operating system. You can get a boot image object by using the **Get-CMBootImage** cmdlet.

Aliases	none
Required?	true

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-BootImageId<String>

Specifies the ID of a boot image.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-BootImageName<String>

Specifies the name of a boot image.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DriverPackage<IResultObject>

Specifies a driver package object. A driver package object specifies a group of hardware drivers that are required to install an operating system. You can get a driver package object by using the **Get-CMDriverPackage** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DriverPackageId<String>

Specifies the ID of a driver package.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DriverPackageName<String>

Specifies the name of a driver package.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OperatingSystemImage<IResultObject>

Specifies an operating system image object. An operating system image object contains the Windows files that compose a complete Windows installation. You can get an operating system image object by using the **Get-CMOperatingSystemImage** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OperatingSystemImageId<String>

Specifies the ID of an operating system image.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OperatingSystemImageName<String>

Specifies the name of an operating system image.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OperatingSystemInstaller<IResultObject>

Specifies an operating system installer object. An operating system installer object contains the Windows files that are required to prepare a computer for the installation of an operating system. To obtain an operating system installer object, use the **Get-CMOperatingSystemInstaller** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OperatingSystemInstallerId<String>

Specifies the ID of an operating system installer.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OperatingSystemInstallerName<String>

Specifies the name of an operating system installer.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Package<IResultObject>

Specifies a deployed software script or program object. You can get a package by using the **Get-CMPackage** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PackageId<String>

Specifies the ID of a deployed software script or program.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PackageName<String>

Specifies the name of a deployed software script or program.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-SoftwareUpdateDeploymentPackage<IResultObject>

Specifies a deployed software update object. You can get a software update deployment object by using the **Get-CMSoftwareUpdateDeploymentPackage** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SoftwareUpdateDeploymentPackageId<String>

Specifies the ID of a deployed software update.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SoftwareUpdateDeploymentPackageName<String>

Specifies the name of a deployed software update.

Aliases	none
Required?	true
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UserName<String>

Specifies a Windows user account name in *domain\user* format.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Modify access to an application by using the application ID

In this example, the first command gets an application ID and stores it in the variable \$ID.

The second command gets the application that is identified by the variable \$ID and adds a user to the access account. The new user's permissions are set to FullControl.

```
PS C:\> $ID = Get-CMAccessAccount -ApplicationID "12994680"  
PS C:\> New-CMAccessAccount -ApplicationID $ID -Type WindowsUser Username  
"CONTOSO\EDaugherty" -Access "FullControl"
```

Related topics

[Get-CMAccessAccount](#)

[Set-CMAccessAccount](#)

[Remove-CMAccessAccount](#)

[Get-CMApplication](#)

[Get-CMBootImage](#)

[Get-CMDriverPackage](#)

[Get-CMOperatingSystemImage](#)

[Get-CMOperatingSystemInstaller](#)

[Get-CMPackage](#)

[Get-CMSoftwareUpdateDeploymentPackage](#)

New-CMAccount

New-CMAccount

Creates a Configuration Manager user account.

Syntax

Parameter Set: NewAccount

```
New-CMAccount -Name <String> -Password <SecureString> -SiteCode <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **New-CMAccount** cmdlet creates a new user account in Microsoft System Center 2012 Configuration Manager. A **CMAccount** is a user account that System Center 2012 Configuration Manager uses to connect to various system and network resources. For more information about user accounts, see [Technical Reference for Accounts Used in Configuration Manager](http://go.microsoft.com/fwlink/?LinkID=248317) (<http://go.microsoft.com/fwlink/?LinkID=248317>) on TechNet.

Parameters

-Name<String>

Specifies a name for the user account.

Aliases	UserName
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Password<SecureString>

Specifies a secure string that contains the password for the user account.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteCode<String>

Specifies a Configuration Manager site code.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Create a new user account by using name and password

The first command creates a password as a secure string.

The second command creates a password confirmation as a secure string.

The third command creates a new account by using the secure strings.

```
PS C:\> $Secure = Read-Host -AsSecureString
PS C:\> $ConfirmSecure = Read-Host -AsSecureString
PS C:\> New-CMAccount -Name "TSQA\PFuller" -Password $Secure -ConfirmPassword
$ConfirmSecure -SiteCode "CM2"
```

Related topics

[Get-CMAccount](#)

[Remove-CMAccount](#)

[Set-CMAccount](#)

New-CMActiveDirectoryForest

New-CMActiveDirectoryForest

Creates one or more Active Directory forest objects in Configuration Manager.

Syntax

Parameter Set: New

```
New-CMActiveDirectoryForest -EnableDiscovery <Boolean> -ForestFqdn <String> [-Description <String> ] [-PublishingPath <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **New-CMActiveDirectoryForest** cmdlet creates an Active Directory forest object that has a fully qualified domain name (FQDN), description, and publishing path that you supply.

If you configured an Active Directory Forest Discovery method, you can enable discovery for an Active Directory forest. After you enable discovery, Microsoft System Center 2012 Configuration Manager discovers Active Directory sites and subnets.

Active Directory Forest Discovery requires a global account to discover or publish to untrusted forests.

Parameters

-Description<String>

Specifies a description for an Active Directory forest object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnableDiscovery<Boolean>

Specifies whether to discover Active Directory sites and subnets. If you enable discovery, you must configure an Active Directory Forest Discovery method. Valid values are \$True or \$False. The default value is \$False.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ForestFqdn<String>

Specifies an FQDN of a Configuration Manager object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PublishingPath<String>

Specifies one or more Configuration Manager sites that publish site information to an Active Directory forest. You can use a comma-separated list in quotation marks to specify more than one site.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Create an Active Directory forest object that has discovery enabled

This command creates an Active Directory forest object that has the FQDN tsqa.contoso.com and that has discovery enabled. You must configure an Active Directory Forest Discovery method before discovery can work.

```
PS C:\> New-CMActiveDirectoryForest -ForestFqdn "tsqa.contoso.com" -EnableDiscovery $True
```

Related topics

[Set-CMActiveDirectoryForest](#)

[Get-CMActiveDirectoryForest](#)

[Remove-CMActiveDirectoryForest](#)

[Get-CMActiveDirectorySite](#)

New-CMAdministrativeUser

New-CMAdministrativeUser

Creates an administrative user for Configuration Manager.

Syntax

Parameter Set: New

```
New-CMAdministrativeUser -Name <String> -RoleName <String[]> [-CollectionName <String[]> ] [-SecurityScopeName <String[]> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **New-CMAdministrativeUser** cmdlet creates an administrative user for Microsoft System Center 2012 Configuration Manager. At the same time that you create the administrative user account, you can give the new administrative user access to collections of System Center 2012 Configuration Manager resources. You can also define the types of access that the new administrative user has to each collection by assigning security roles to the user.

For more information about adding administrative users, see [Configuring Security for Configuration Manager](http://go.microsoft.com/fwlink/?linkid=247674) (http://go.microsoft.com/fwlink/?linkid=247674) on TechNet.

Parameters

-CollectionName<String[]>

Specifies an array of collection names. The cmdlet assigns the new administrative user to each of these collections.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies the name of the administrative user in the form <domain>\<user>.

Aliases	LogonName
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	true

-RoleName<String[]>

Specifies an array of names for the roles that you assign to an administrative user. Valid values are:

- Application Administrator
- Application Author
- Application Deployment Manager
- Asset Manager
- Compliance Settings Manager
- Discovery Operator
- Endpoint Protection Manager
- Full Administrator
- Infrastructure Administrator
- Operating System Deployment Manager
- Operations Administrator
- Read-only Analyst
- Remote Tools Operator
- Security Administrator
- Software Update Manager
- Custom-defined security roles

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-SecurityScopeName<String[]>

Specifies an array of names of security scopes. A security scope name can be Default or the name of a custom security scope. The cmdlet assigns the security scopes that you specify to the administrative user.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Creates an administrative user

This command adds the user named pattifuller on the NorthAmerica domain as an administrative user with the Application Author security role. The command adds this user to the resource collection named ClientAdminResources.

```
PS C:\> New-CMAdministrativeUser -Name "NorthAmerica\pattifuller" -CollectionName  
"ClientAdminResources" -RoleName "Application Author"
```

Related topics

[Get-CMAdministrativeUser](#)

[Remove-CMAdministrativeUser](#)

New-CMAAlertSubscription

New-CMAAlertSubscription

Creates an alert subscription object.

Syntax

Parameter Set: New

```
New-CMAAlertSubscription -EmailAddress <String[]> -Name <String> [-AlertId <Int32[]> ] [-LocaleId <Int32> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **New-CMAAlertSubscription** cmdlet creates a subscription that sends alert notifications to one or more users when specific events occur in Microsoft System Center 2012 Configuration Manager. Before you create an alert subscription, make sure that you have configured email settings for sending alert notifications, and that you have at least one alert configured in System Center 2012 Configuration Manager.

Parameters

-AlertId<Int32[]>

Specifies an array of alert identifiers for the subscription.

Aliases	AlertIds
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EmailAddress<String[]>

Specifies an email address where you want to send an alert notification. For example, david.chew@contoso.com. You can separate multiple email addresses by using a semicolon.

Aliases	EmailAddresses
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-LocaleId<Int32>

Specifies a locale for alert messages. For more information and a list of locale identifiers, see the [Locale IDs Assigned by Microsoft](http://go.microsoft.com/fwlink/?LinkId=262651) topic at <http://go.microsoft.com/fwlink/?LinkId=262651>.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies the name of an alert subscription object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Create a new alert subscription

This command creates an alert subscription named Subscription01 and sends alert notifications to a specific email recipient in English whenever there is an event that pertains to a specified alert.

```
PS C:\> New-CMAAlertSubscription -Name "Subscription01" -EmailAddress  
"evan.narvaez@contoso.com" -LocaleId 1033 -AlertIds 16777219
```

Related topics

[Get-CMAAlertSubscription](#)

[Set-CMAAlertSubscription](#)

[Remove-CMAAlertSubscription](#)

New-CMAmtProvisioningAccount

New-CMAmtProvisioningAccount

Creates an AMT Discovery and Provisioning Account.

Syntax

Parameter Set: NewAmtProvisioningAccount

```
New-CMAmtProvisioningAccount -Password <SecureString> -UserName <String> [-Description <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **New-CMAmtProvisioningAccount** cmdlet creates an account for AMT provisioning and discovery for the Intel Active Management Technology (Intel AMT)-based computers that Microsoft System Center 2012 Configuration Manager manages out of band.

The server that runs the out of band service point role uses this account to manage some network interface features of AMT in System Center 2012 Configuration Manager, by using the out of band management feature. The AMT Provisioning and Discovery Account that you specify in System Center 2012 Configuration Manager must match the AMT Remote Admin Account name and password of the BIOS extensions in the AMT-based computers.

Parameters

-Description<String>

Specifies a description for the AMT Discovery and Provisioning Account.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Password<SecureString>

Specifies the password, as a secure string, for the AMT Discovery and Provisioning Account.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UserName<String>

Specifies the name of the MEBx Account or Remote Admin Account of the BIOS extensions in the AMT-based computers.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Create an AMT Discovery and Provisioning Account

This command creates an AMT Discovery and Provisioning Account named AMT_Manager, and specifies a password and description for the account.

```
PS C:\> New-CMAmtProvisioningAccount -Username "AMT_Manager" -Password "S@mPle1Pswrd" -Description "Out-of-band management security group"
```

Related topics

[Enable-CMAutomaticAMTProvisioning](#)

[Invoke-CMAmtProvisioningDiscovery](#)

New-CMAntimalwarePolicy

New-CMAntimalwarePolicy

Creates an object that specifies antimalware policies.

Syntax

Parameter Set: New

```
New-CMAntimalwarePolicy -Name <String> -Policy {Advanced | DefaultActions |  
DefinitionUpdates | ExclusionSettings | MicrosoftActiveProtectionService |  
RealTimeProtection | ScanSettings | ScheduledScans | ThreatOverrides} [-Description <String>  
] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **New-CMAntimalwarePolicy** cmdlet creates an object that specifies the antimalware policies for System Center 2012 Endpoint Protection that apply to collections of client computers that run a Microsoft System Center 2012 Configuration Manager agent. Antimalware policies are configuration settings that define how an antimalware agent operates on a client computer.

Parameters

-Description<String>

Specifies a description for the policy.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies a name for the policy.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Policy<PolicyType[]>

Specifies an array of antimalware policy types. Valid values are:

- Advanced
- DefaultActions
- DefinitionUpdates
- ExclusionSettings
- MicrosoftActiveProtectionService
- RealTimeProtection
- ScanSettings
- ScheduledScans
- ThreatOverrides

The acceptable values for this parameter are:

Advanced	
DefaultActions	
DefinitionUpdates	
ExclusionSettings	
MicrosoftActiveProtectionService	
RealTimeProtection	
ScanSettings	
ScheduledScans	
ThreatOverrides	

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Create an antimalware policy

This command creates an antimalware policy named ExclusionsPolicy. This policy has the policy type of ExclusionSettings. The command includes an optional description that describes the policy.

```
PS C:\> New-CMAntimalwarePolicy -Name "ExclusionsPolicy" -Policy ExclusionSettings -  
Description "Policy defining exclusions."
```

Related topics

[Export-CMAntimalwarePolicy](#)

[Get-CMAntiMalwarePolicy](#)

[Merge-CMAntimalwarePolicy](#)

[Remove-CMAntiMalwarePolicy](#)

[Set-CMAntiMalwarePolicy](#)

[Start-CMAntimalwarePolicyDeployment](#)

New-CMApplication

New-CMApplication

Creates an application in Configuration Manager.

Syntax

Parameter Set: New

```
New-CMApplication -Name <String> [-AutoInstall <Boolean> ] [-Description <String> ] [-IconLocationFile <String> ] [-IsFeatured <Boolean> ] [-Keyword <String> ] [-LinkText <String> ] [-LocalizedApplicationDescription <String> ] [-LocalizedApplicationName <String> ] [-OptionalReference <String> ] [-Owner <String> ] [-PrivacyUrl <String> ] [-Publisher <String> ] [-ReleaseDate <DateTime> ] [-SoftwareVersion <String> ] [-SupportContact <String> ] [-UserDocumentation <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **New-CMApplication** cmdlet creates a new application in Microsoft System Center 2012 Configuration Manager.

Parameters

-AutoInstall<Boolean>

Specifies whether a task sequence action can install the application.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Description<String>

Specifies a description for the application. The description appears in the administrator console.

Aliases	LocalizedDescription
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-IconLocationFile<String>

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-IsFeatured<Boolean>

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Keyword<String>

Specifies a keyword for the application.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-LinkText<String>

Specifies a description that appears in the Application Catalog with the hyperlink to additional information or documentation for the application.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-LocalizedApplicationDescription<String>

Specifies a localized description string that appears in the client software center or catalog web site.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-LocalizedApplicationName<String>

Specifies a localized name string that appears in the client software center or catalog web site.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies a name for the application.

Aliases	LocalizedDisplayName
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OptionalReference<String>

Specifies optional reference information for this application.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Owner<String>

Specifies an owner for the application.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PrivacyUrl<String>

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Publisher<String>

Specifies the name of a software publisher in Configuration Manager.

Aliases	Manufacturer
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ReleaseDate<DateTime>

Specifies a release date for the application.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SoftwareVersion<String>

Specifies a software version for the application.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SupportContact<String>

Specifies one or more administrative users who are support contacts for the application.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UserDocumentation<String>

Specifies a hyperlink, in URI format, to additional information about the application.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Create an application

This command creates a new application.

```
PS C:\> New-CMApplication -Name "Contoso-test" -AutoInstall $True -Description "Contoso-  
test1" -Keyword "Contoso" -LinkText "Contoso-App" -LocalizedApplicationDescription "Contoso-  
test" -LocalizedApplicationName "Contoso-test" -Manufacturer "Contoso-Mfg" -  
OptionalReference "Contoso-test4" -Owner "Contoso-testOwner" -ReleaseDate 2013/5/24 -  
SoftwareVersion "v3" -SupportContact "Contoso-supp" -UserDocumentation "Contoso-Help:"
```

Related topics

[Export-CMApplication](#)

[Get-CMApplication](#)

[Import-CMApplication](#)

[Remove-CMApplication](#)

[Resume-CMApplication](#)

[Set-CMApplication](#)

[Suspend-CMApplication](#)

New-CMAppVVirtualEnvironment

New-CMAppVVirtualEnvironment

Creates an App-V virtual environment.

Syntax

Parameter Set: New

```
New-CMAppVVirtualEnvironment -ApplicationGroup <VirtualEnvironmentGroup[]> -Name <String> [-Description <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **New-CMAppVVirtualEnvironment** cmdlet creates an Microsoft Application Virtualization (App-V) virtual environment in Microsoft System Center 2012 Configuration Manager. App-V virtual environments in System Center 2012 Configuration Manager enable deployed virtual applications to share the same file system and registry on client computers.

Parameters

-ApplicationGroup<VirtualEnvironmentGroup[]>

Specifies an array of application groups to add to the App-V virtual environment. To obtain an application group, use the **New-CMVirtualEnvironmentGroup** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Description<String>

Specifies a description for the App-V virtual environment.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies a name for the App-V virtual environment.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Create an App-V virtual environment

This first command uses the **Get-CMAppV5XDeploymentTypeItem** cmdlet gets the deployment type named 7Zip - Microsoft Application Virtualization 5 in the application named App01d2012. The command stores the result in the \$Dti variable.

This second command creates an application group named Venvgroup01 for the deployment type stored in \$Dti. The command stores the result in the \$Veg variable.

This third command creates an App-V virtual environment named CMAppVenv01 for the application group stored in \$Veg.

```
PS C:\> $Dti = Get-CMAppV5XDeploymentTypeItem -ApplicationName "App01d2012" -
DeploymentTypeName "7Zip - Microsoft Application Virtualization 5"
PS C:\> $Veg = New-CMVirtualEnvironmentGroup -Name "Venvgroup01" -DeploymentType $Dti
PS C:\> New-CMAppVVirtualEnvironment -Name "CMAppVenv01" -Description "App-V virtual
environment" -ApplicationGroup $Veg
```

Related topics

[Get-CMAppVVirtualEnvironment](#)

[Remove-CMAppVVirtualEnvironment](#)

[Set-CMAppVVirtualEnvironment](#)

[Get-CMAppV5XDeploymentTypeItem](#)

[New-CMVirtualEnvironmentGroup](#)

New-CMAssetIntelligenceCatalogItem

New-CMAssetIntelligenceCatalogItem

Creates an item for the Asset Intelligence catalog.

Syntax

Parameter Set: New

```
New-CMAssetIntelligenceCatalogItem -CategoryName <String> -Type {TypeCategory | TypeFamily | TypeTag} [-Description <String> ] [-LanguageId <Int32> ] [-Confirm] [-WhatIf] [<CommonParameters>]
```

Detailed Description

The **New-CMAssetIntelligenceCatalogItem** cmdlet creates software categories, software families, and custom software labels from the Asset Intelligence catalog in Microsoft System Center 2012 Configuration Manager.

The Asset Intelligence catalog contains categorization and identification information for software titles. The catalog includes predefined categories and families. Predefined items cannot be modified. In addition to predefined software categories and software families, you can create custom categories and families. You can also create custom software labels.

Parameters

-CategoryName<String>

Specifies the name of a category, family, or label in the Asset Intelligence catalog.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Description<String>

Specifies the description of a category, family, or label in the Asset Intelligence catalog.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-LanguageId<Int32>

Specifies the locale identifier for an item. For more information and a list of locale identifiers, see the [Locale IDs Assigned by Microsoft](http://go.microsoft.com/fwlink/?LinkId=262651) topic at <http://go.microsoft.com/fwlink/?LinkId=262651>.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Type<Types>

Specifies the type of Asset Intelligence catalog item. Possible values are TYPE_CATEGORY, TYPE_FAMILY, and TYPE_TAG.

The acceptable values for this parameter are:

TypeCategory	
TypeFamily	
TypeTag	

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Create category label item in the catalog

This command creates a category label in the Asset Intelligence catalog named Databases that has a language ID of 1033 and a type of TYPE_TAG.

```
PS C:\> New-CMAAssetIntelligenceCatalogItem -CategoryName "Databases" -LanguageId 1033 -Type TYPE_TAG
```

Example 2: Create a category item in the catalog

This command creates a category in the Asset Intelligence catalog named Database Tools that has a type of TYPE_CATEGORY.

```
PS C:\> New-CMAAssetIntelligenceCatalogItem -CategoryName "Database Tools" -Type TYPE_CATEGORY
```

Example 3: Create a category family in the catalog

This command creates a category family item in the Asset Intelligence catalog family named Database Software that has a type of TYPE_FAMILY.

```
PS C:\> New-CMAAssetIntelligenceCatalogItem -CategoryName "Database Software" -Type TYPE_FAMILY
```

Related topics

[Get-CMAAssetIntelligenceCatalogItem](#)

[Set-CMAAssetIntelligenceCatalogItem](#)

[Remove-CMAAssetIntelligenceCatalogItem](#)

New-CMBaseline

New-CMBaseline

Creates a Configuration Manager baseline.

Syntax

Parameter Set: New

```
New-CMBaseline -Name <String> [-Category <String[]> ] [-Description <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **New-CMBaseline** cmdlet creates a Microsoft System Center 2012 Configuration Manager baseline. A baseline is a collection of configuration items that System Center 2012 Configuration Manager uses to evaluate whether a computer complies with software requirements. After you create a baseline, you can deploy it to a collection so that devices in that collection download the configuration baseline and assess compliance with it.

Parameters

-Category<String[]>

Specifies an array of categories to which the baseline configuration belongs. Valid values are:

- Client
- IT Infrastructure
- Line of Business
- Server

Aliases	LocalizedCategoryInstanceNames
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Description<String>

Specifies a description for the baseline.

Aliases	LocalizedDescription
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies a name for the configuration baseline.

Aliases	LocalizedDisplayName
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Create a configuration baseline

This command creates a baseline for compliance named Accounting Department baseline. The command specifies a description for the baseline.

```
PS C:\> New-CMBaseline -Name "Accounting Department baseline" -Description "Compliance standards for Accounting computers."
```

Related topics

[Disable-CMBaseline](#)

[Enable-CMBaseline](#)

[Export-CMBaseline](#)

[Get-CMBaseline](#)

[Import-CMBaseline](#)

[Remove-CMBaseline](#)

[Set-CMBaseline](#)

New-CMBootImage

New-CMBootImage

Adds a new operating system boot image.

Syntax

Parameter Set: New

```
New-CMBootImage -Index <Int32> -Name <String> -Path <String> [-Description <String> ] [-Version <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **New-CMBootImage** cmdlet adds a new Windows Preinstallation Environment (Windows PE) operating system boot image to Microsoft System Center 2012 Configuration Manager. Operating system images are .wim format files. These files contain a compressed set of reference files and folders that are required to successfully install and configure a boot image on a computer. By default, System Center 2012 Configuration Manager includes both x86 and x64 operating system images.

You must run **New-CMBootImage** on the computer that is running the Systems Management Server (SMS) provider. The computer account of the computer that is running the SMS provider must have Read and Write access to the package source of the boot image. For more information about the SMS provider, see [Planning for the SMS Provider in Configuration Manager](http://go.microsoft.com/fwlink/?LinkID=263566) (<http://go.microsoft.com/fwlink/?LinkID=263566>) on TechNet.

Parameters

-Description<String>

Specifies a description of the boot image.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Index<Int32>

Specifies an index. An index is the number of an image in a .wim file.

Aliases	ImageIndex
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies the name of a new boot image.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Path<String>

Specifies the location of the original WIM file for the boot image.

Aliases	ImagePath
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Version<String>

Specifies the version of the boot image.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Create a new boot image object

This command creates a new boot image object and provides it with a source path for its WIM file, an index value, a name, an operating system version, and a description.

```
PS C:\> New-CMBootImage -Path "\\Server99.Contoso.com\SMS_CCC\osd\boot\i386\boot.wim" -Index 1 -Name "WinPE Boot Image" -Version 11 -Description "WinPE Boot Image x86 Approved 9/1/2012"
```

Related topics

[Get-CMBootImage](#)

[Remove-CMBootImage](#)

[Set-CMBootImage](#)

New-CMBoundary

New-CMBoundary

Creates a new boundary.

Syntax

Parameter Set: New

```
New-CMBoundary -Type {ADSite | IPRange | IPSubnet | IPV6Prefix} -Value <String> [-Name <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **New-CMBoundary** cmdlet creates a new boundary.

In Microsoft System Center 2012 Configuration Manager, a boundary is an intranet location that contains one or more devices that you can manage. A boundary can be an IP subnet, Active Directory site name, IPv6 prefix, or an IP address range.

Parameters

-Name<String>

Specifies the name of the new boundary.

Aliases	DisplayName
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Type<BoundaryTypes>

Specifies a boundary type. Valid values are: ADSite, IPV6Prefix, IPSubnet, and IPRange.

The acceptable values for this parameter are:

ADSite	
IPRange	
IPSubnet	
IPV6Prefix	

Aliases	BoundaryType
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Value<String>

Specifies the data that describes the boundary. For example, an Active Directory site value can be Default-First-Site-Name.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Create a new IP Subnet site boundary

This command creates a new IP subnet site boundary that has a name of IPSubNetBoundary01 and a value of 172.16.50.0/24.

```
PS C:\> New-CMBoundary -DisplayName "IPSubNetBoundary01" -BoundaryType IPSubNet -Value "172.16.50.0/24"
```

```
BoundaryFlags:          0
BoundaryID:            6338009
BoundaryType:          0
CreatedBy:              Contoso\PFuller
CreatedOn:              6/10/2012 1:17:42 PM
DefaultSiteCode:
DisplayName:            IPSubNetBoundary01
GroupCount:            0
ModifiedBy:
ModifiedOn:
SiteSystems:
```

Value: 172.16.50.0/24

Example 2: Create a new Active Directory site boundary

This command creates a new Active Directory site boundary that has a name of ADSiteBoundary01 and a value of Default-First-Site-Name.

```
PS C:\> New-CMBoundary -DisplayName "ADSiteBoundary01" -BoundaryType ADSite -Value "Default-First-Site-Name"
```

```
BoundaryFlags: 0
BoundaryID: 6339999
BoundaryType: 1
CreatedBy: Contoso\PFuller
CreatedOn: 6/10/2012 2:58:56 PM
DefaultSiteCode:
DisplayName: ADSiteBoundary01
GroupCount: 0
ModifiedBy:
SiteSystems:
Value: Default-First-Site-Name
```

Example 3: Create a new IP v6 prefix site boundary

This command creates a new IP v6 prefix site boundary that has a name of IPv6PrefixBoundary01 and a value of FE80::/64.

```
PS C:\> New-CMBoundary -DisplayName "IPv6PrefixBoundary01" -BoundaryType IPv6Prefix -Value "FE80::/64".
```

```
BoundaryFlags: 0
BoundaryID: 63347110
BoundaryType: 2
CreatedBy: Contoso\PFuller
CreatedOn: 6/10/2012 3:15:19 PM
DefaultSiteCode:
DisplayName: IPv6PrefixBoundary01
GroupCount: 0
ModifiedBy:
ModifiedOn:
SiteSystems:
Value: "FE80::/64"
```

Example 4: Create a new IP range site boundary

This command creates a new IP range site boundary that has the name IPRangeBoundary01 and a value of 10.255.255.0-10.255.255.255.

```
PS C:\> New-CMBoundary -DisplayName "IPRangeBoundary01" -BoundaryType IPRange -Value "10.255.255.0-10.255.255.255"
```

```
BoundaryFlags:      0
BoundaryID:         6334129
BoundaryType:       3
CreatedBy:          Contoso\PFuller
CreatedOn:          6/10/2012 3:29:05 PM
DefaultSiteCode:
DisplayName:        IPRangeBoundary01
GroupCount:         0
ModifiedBy:
ModifiedOn:
SiteSystems:
Value:              10.255.255.0-10.255.255.255
```

Related topics

[Get-CMBoundary](#)

[Remove-CMBoundary](#)

[Set-CMBoundary](#)

New-CMBoundaryGroup

New-CMBoundaryGroup

Creates a new boundary group.

Syntax

Parameter Set: New

```
New-CMBoundaryGroup -Name <String> [-DefaultSiteCode <String> ] [-Description <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **New-CMBoundaryGroup** cmdlet creates a new boundary group. A boundary group is a collection of boundaries.

You can use boundary groups to manage network locations. You must assign boundaries to boundary groups before you can use the boundary group. Boundary groups enable client computers to find a primary site for client assignment, which is referred to as automatic site assignment, and a list of available site systems that have content. For more information about boundaries, see [Planning for Boundaries and Boundary Groups in Configuration Manager](#) (<http://go.microsoft.com/fwlink/?LinkId=266225>) on TechNet and the **New-CMBoundary** cmdlet.

Parameters

-DefaultSiteCode<String>

Specifies the default site code for the boundary group.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Description<String>

Specifies a description for the new boundary.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies a name for the new boundary.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Create a new boundary group

This command creates a new boundary group. After the new boundary group is created, the command displays an unpopulated list of boundary properties. To refresh and see a populated list, use the **Get-CMBoundaryGroup** cmdlet. The output shown for this example is the latter.

```
PS C:\> New-BoundaryGroup -Name "BGroup05"
```

```
CreatedBy:
CreatedOn:
DefaultSiteCode:
Description:
GroupID:
MemberCount:          0
ModifiedBy:
ModifiedOn:
Name:                 BGroup05
SiteSystemCount:
```

Related topics

[Get-CMBoundaryGroup](#)

[Remove-CMBoundaryGroup](#)

[Set-CMBoundaryGroup](#)

[New-CMBoundary](#)

New-CMCategory

New-CMCategory

Creates a configuration category in Configuration Manager.

Syntax

Parameter Set: NewCategory

```
New-CMCategory -CategoryType {UserCategories | BaselineCategories | DriverCategories | AppCategories | GlobalCondition | CatalogCategories} -Name <String> [-Confirm] [-WhatIf] [<CommonParameters>]
```

Detailed Description

The **New-CMCategory** cmdlet creates a configuration category in Microsoft System Center 2012 Configuration Manager. Configuration categories offer an optional method of sorting and filtering configuration baselines and configuration items in System Center 2012 Configuration Manager and Configuration Manager reports.

Parameters

-CategoryType<CategoryType>

Specifies a category type. Valid values are:

- BaselineCategories
- DriverCategories
- AppCategories
- GlobalCondition
- CatalogCategories

The acceptable values for this parameter are:

UserCategories	
BaselineCategories	
DriverCategories	
AppCategories	

GlobalCondition	
CatalogCategories	

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies a name for the configuration category.

Aliases	LocalizedCategoryInstanceName
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Create a configuration category

This command creates a new category in DriverCategories named NewLaptopDriverSet.

```
PS C:\> New-CMCategory -CategoryType "DriverCategories" -Name "NewLaptopDriverSet"
```

Related topics

[Remove-CMCategory](#)

[Get-CMCategory](#)

New- CMClientAuthCertificateProfileConfigurationItem

New-CMClientAuthCertificateProfileConfigurationItem

Creates a certificate profile.

Syntax

Parameter Set: Default

```
New-CMClientAuthCertificateProfileConfigurationItem -DesiredConfigurationDigestPath <String>  
[-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **New-CMClientAuthCertificateProfileConfigurationItem** cmdlet creates a certificate profile. Client computers use certificate profiles to authenticate when they use services such as a virtual private network (VPN) or a wireless network.

Parameters

-DesiredConfigurationDigestPath<String>

Specifies a path to the configuration data stored as a digest.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

[Copy-CMClientAuthCertificateProfileConfigurationItem](#)

[Get-CMClientAuthCertificateProfileConfigurationItem](#)

[Remove-CMClientAuthCertificateProfileConfigurationItem](#)

[Set-CMClientAuthCertificateProfileConfigurationItem](#)

New-CMClientSetting

New-CMClientSetting

Creates customized client settings.

Syntax

Parameter Set: New

```
New-CMClientSetting -Name <String> [-Description <String> ] [-Type {Default | Device | User}  
] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **New-CMClientSetting** cmdlet creates a collection of customized settings for Microsoft System Center 2012 Configuration Manager client computers. After you create the customized settings and deploy them to client computer collections, the customized settings override the default client settings for that collection.

For more information about client settings, see [About Client Settings in Configuration Manager](http://go.microsoft.com/fwlink/?LinkId=266226) (http://go.microsoft.com/fwlink/?LinkId=266226) on TechNet.

Parameters

-Description<String>

Specifies a description of the content of the new settings.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies a name for customized client settings.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Type<Types>

Specifies the type of customized settings. Valid values are: 1 (device) or 2 (user).

The acceptable values for this parameter are:

Default	
Device	
User	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Create a customized collection of client settings

This command creates customized device settings for the group of client computers that run Windows® 8. After the new collection of settings is created, the command displays an unpopulated list of setting properties. To refresh and see a populated list of properties, use **Get-CMClientSetting**. The output for this example shows a populated list.

```
PS C:\> New-CMClientSetting -Name "Win08ClientSettings" -Description "Windows 8 Client Computers Settings" -Type 1
```

```
AgentConfigurations: {}
AssignmentCount:     0
CreatedBy:           Contoso\DChew
DateCreated:         8/04/2012 4:40:03 PM
DateModified:        8/04/2012 4:40:03 PM
Description:         Windows 8 Client Computers Settings
Enabled:             False
FeatureType:         1
```

Flags: 0
LastModifiedBy: Contoso\DChew
Name: Win08ClientSettings
Priority: 0
SecuredScopeNames: {Default}
Settings ID: 16777220
Type: 1
UniqueID: {0CCA6700-AE5E-4949-8FBC-AA6719775CC3}

Related topics

[Get-CMClientSetting](#)

[Remove-CMClientSetting](#)

[Set-CMClientSetting](#)

New-CMCloudDistributionPoint

New-CMCloudDistributionPoint

Creates a cloud distribution point.

Syntax

Parameter Set: NewCloudDistributionPoint

```
New-CMCloudDistributionPoint -ManagementCertificatePath <String> -Region {AnywhereAsia | AnywhereEurope | AnywhereUS | EastAsia | EastUS | NorthCentralUS | NorthEurope | SouthCentralUS | SoutheastAsia | WestEurope | WestUS} -ServiceCertificatePath <String> -ServiceCName <String> -SiteCode <String> -SubscriptionId <String> [-Description <String> ] [-ManagementCertificatePassword <SecureString> ] [-ServiceCertificatePassword <SecureString> ] [-StorageCriticalThreshold <Int32> ] [-StorageQuotaGB <Int32> ] [-StorageWarningThreshold <Int32> ] [-TrafficCriticalThreshold <Int32> ] [-TrafficOutGB <Int32> ] [-TrafficWarningThreshold <Int32> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **New-CMCloudDistributionPoint** cmdlet creates a cloud distribution point in Microsoft System Center 2012 Configuration Manager.

In System Center 2012 Configuration Manager, you can use a cloud service in Windows Azure to host a distribution point for storing files to download to clients. You can send packages and apps to and host packages and apps in cloud distribution points. For more information about cloud distribution points, see [Planning for Content Management in Configuration Manager](http://go.microsoft.com/fwlink/?LinkId=266223) (<http://go.microsoft.com/fwlink/?LinkId=266223>) on TechNet.

Parameters

-Description<String>

Specifies a description for a cloud distribution point.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ManagementCertificatePassword<SecureString>

Specifies a password for a management certificate.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ManagementCertificatePath<String>

Specifies a location for a management certificate.

Aliases	ManagementCertificate
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Region<Region>

Specifies a region. This is the Windows Azure region where you want to create the cloud service that hosts this distribution point. Valid values are:

- AnywhereAsia
- AnywhereEurope
- AnywhereUS
- EastAsia
- EastUS

- NorthCentralUS
- NorthEurope
- SouthCentralUS
- SoutheastAsia
- WestEurope
- WestUS

The acceptable values for this parameter are:

AnywhereAsia	
AnywhereEurope	
AnywhereUS	
EastAsia	
EastUS	
NorthCentralUS	
NorthEurope	
SouthCentralUS	
SoutheastAsia	
WestEurope	
WestUS	

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ServiceCertificatePassword<SecureString>

Specifies a password for a service certificate.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ServiceCertificatePath<String>

Specifies a location for a service certificate.

Aliases	ServiceCertificate
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ServiceCName<String>

Specifies an alias, or CName, for a service.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteCode<String>

Specifies a Configuration Manager site code.

Aliases	none
---------	------

Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-StorageCriticalThreshold<Int32>

Specifies the percentage for a critical alert to occur, based on the storage alert threshold.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-StorageQuotaGB<Int32>

Specifies the threshold value, in gigabytes, that triggers errors or warnings for total content storage.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-StorageWarningThreshold<Int32>

Specifies the percentage for a warning alert to occur, based on the storage alert threshold.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SubscriptionId<String>

Specifies a subscription ID for your Windows Azure account. To obtain a subscription ID, use the Windows Azure Management Portal.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-TrafficCriticalThreshold<Int32>

Specifies the percentage for a critical alert to occur, based on the traffic out alert threshold.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-TrafficOutGB<Int32>

Specifies the threshold value, in gigabytes, that triggers errors or warnings, for monthly traffic out of Windows Azure Storage Service.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-TrafficWarningThreshold<Int32>

Specifies the percentage for a warning alert to occur, based on the traffic out alert threshold.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

[Get-CMCloudDistributionPoint](#)

[Remove-CMCloudDistributionPoint](#)

[Set-CMCloudDistributionPoint](#)

[Start-CMCloudDistributionPoint](#)

[Stop-CMCloudDistributionPoint](#)

New-CMComputerAssociation

New-CMComputerAssociation

Creates an association between two computers in Configuration Manager.

Syntax

Parameter Set: NewComputerAssociation

```
New-CMComputerAssociation -DestinationComputer <String> -SourceComputer <String> [-MigrationBehavior {CaptureAllUserAccountsAndRestoreSpecifiedAccounts | CaptureAndRestoreAllUserAccounts | CaptureAndRestoreSpecifiedUserAccounts} ] [-MigrationUserName <String[]> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **New-CMComputerAssociation** cmdlet creates an association between two computers to use for migration. Microsoft System Center 2012 Configuration Manager can migrate user state and settings from an existing computer to a different computer as part of operating system deployment. In the course of migration, System Center 2012 Configuration Manager saves accounts created on the source computer and creates those user accounts on the destination computer.

To create an association, specify the source computer, the destination computer, and at least one user name created on the source computer to be migrated. You can also specify whether the migration includes other user names from the source computer.

Parameters

-DestinationComputer<String>

Specifies the name of a destination computer.

Aliases	RestoreName
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-MigrationBehavior<MigrationBehavior>

Specifies how Configuration Manager treats user accounts created on the source computer. When you create a computer association, specify user accounts created on the source computer by using the *MigrationUserName* parameter. The computer association can specify that the migration process creates some or all of those accounts on the destination computer.

Valid values are:

-- CaptureAllUserAccountsAndRestoreSpecifiedAccounts. Saves all accounts created on the source computer, but creates only the specified accounts on the destination computer.

-- CaptureAndRestoreAllUserAccounts. Saves all accounts created on the source computer, and creates them on the destination computer.

-- CaptureAndRestoreSpecifiedUserAccounts. Saves only the specified accounts from the source computer, and creates those accounts on the destination computer.

If you do not specify a migration behavior, the migration uses CaptureAndRestoreAllUserAccounts.

The acceptable values for this parameter are:

CaptureAllUserAccountsAndRestoreSpecifiedAccounts	
CaptureAndRestoreAllUserAccounts	
CaptureAndRestoreSpecifiedUserAccounts	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-MigrationUserName<String[]>

Specifies an array of user names for accounts created on the source computer. The specified user names, along with the *MigrationBehavior* parameter setting, determine which user accounts Configuration Manager creates on the destination computer.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SourceComputer<String>

Specifies the name of the source computer.

Aliases	SourceName
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Create a computer association

This command creates a computer association between the source computer named TSQA073 and the destination computer named TSQA155. The command specifies a user name for migration to the destination computer.

```
PS C:\> New-CMComputerAssociation -SourceComputer "TSQA073" -MigrationUserName "Contoso-TSQA\ElisaDaugherty" -DestinationComputer "TSQA155"
```

Related topics

[Get-CMComputerAssociation](#)

[Remove-CMComputerAssociation](#)

[Set-CMComputerAssociation](#)

New-CMConfigurationItem

New-CMConfigurationItem

Creates a configuration item.

Syntax

Parameter Set: New

```
New-CMConfigurationItem -CreationType {MacOS | MobileDevice | None | WindowsApplication | WindowsOS} -Name <String> [-Category <String[]> ] [-Description <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: NewChild

```
New-CMConfigurationItem -Name <String> -ParentConfigurationItem <IResultObject> [-Category <String[]> ] [-Description <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **New-CMConfigurationItem** cmdlet creates a configuration item in Microsoft System Center 2012 Configuration Manager. Create configuration items to define configurations that you want to manage and assess for compliance on devices.

You can specify the *ParentConfigurationItem* parameter to create a child configuration item. Child configuration items in System Center 2012 Configuration Manager are copies of configuration items that retain a relationship to the original configuration item; therefore, they inherit the original configuration from the parent configuration item. You cannot create child configuration items for mobile devices.

Parameters

-Category<String[]>

Specifies an array of localized names of the categories to which the configuration item belongs.

Aliases	LocalizedCategoryInstanceNames
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CreationType<CICreationType>

Specifies the type of configuration item. Valid values are:

- MacOS
- MobileDevice
- None
- WindowsApplication
- WindowsOS

The acceptable values for this parameter are:

MacOS	
MobileDevice	
None	
WindowsApplication	
WindowsOS	

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Description<String>

Specifies a description for a configuration item.

Aliases	LocalizedDescription
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies a name for the configuration item.

Aliases	LocalizedDisplayName
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ParentConfigurationItem<IResultObject>

Specifies a parent **CMConfigurationItem** object. To obtain a **CMConfigurationItem** object, use the [Get-CMConfigurationItem](#) cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Create a configuration item

This command creates a configuration item for mobile devices named MD_Config88.

```
PS C:\> New-CMConfigurationItem -CreationType MobileDevice -Name "MD_Config88"
```

Related topics

[Get-CMConfigurationItem](#)

[Set-CMConfigurationItem](#)

[Import-CMConfigurationItem](#)

[Remove-CMConfigurationItem](#)

[Export-CMConfigurationItem](#)

New-CMDeviceCollection

New-CMDeviceCollection

Creates a collection for devices and adds the collection to the Configuration Manager hierarchy.

Syntax

Parameter Set: NewByLimitName

```
New-CMDeviceCollection -LimitingCollectionName <String> -Name <String> [-Comment <String> ]  
[-RefreshSchedule <IResultObject> ] [-RefreshType {Both | ConstantUpdate | Manual |  
Periodic} ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: NewByLimitId

```
New-CMDeviceCollection -LimitingCollectionId <String> -Name <String> [-Comment <String> ] [-  
RefreshSchedule <IResultObject> ] [-RefreshType {Both | ConstantUpdate | Manual | Periodic}  
] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **New-CMDeviceCollection** cmdlet creates a collection based on a specific limiting collection. The limiting collection determines which devices can be a member of the device collection that you create. For example, when you use the All Systems collection as the limiting collection, the new collection can include any device in the Configuration Manager hierarchy. You specify the limiting collection by providing its name or ID.

Devices are added to the collection by membership rules. To add members to the device collection use one of the following membership rule cmdlets:

- [Add-CMDeviceCollectionDirectMembershipRule](#)
- [Add-CMDeviceCollectionExcludeMembershipRule](#)
- [Add-CMDeviceCollectionIncludeMembershipRule](#)
- [Add-CMDeviceCollectionQueryMembershipRule](#)

Collections represent logical groupings of resources, such as users and devices. For more information about Configuration Manager collections, see [Introduction to Collections in Configuration Manager](http://go.microsoft.com/fwlink/p/?LinkID=259433) (<http://go.microsoft.com/fwlink/p/?LinkID=259433>) on TechNet.

Parameters

-Comment<String>

Specifies a description of the collection, such as what type of devices are included in the collection.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-LimitingCollectionId<String>

Specifies the ID of the collection that Configuration Manager uses to limit which devices are available to the collection that you are creating. For example, the following ID is the ID of the All Systems collection: SMS00001.

Aliases	LimitToCollectionId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-LimitingCollectionName<String>

Specifies the name of the collection that Configuration Manager uses to limit which devices are available to the collection that you are creating. For example, you can specify the All Systems collection.

Aliases	LimitToCollectionName
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies the name for the collection that you want to create. The collection name might reflect an organization or what type of devices are in the collection. For example, you could create a Windows 7 collection to contain all devices running Windows 7.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RefreshSchedule<IRResultObject>

Specifies a schedule that determines when Configuration Manager refreshes the collection.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RefreshType<RefreshTypes>

Specifies how Configuration Manager refreshes the collection. Valid values are:

-- Manual. The collection is refreshed using the Configuration Manager console or the Configuration Manager SDK.

-- Periodic. The collection is refreshed based on the schedule specified by the RefreshSchedule parameter.

-- ConstantUpdate. The collection is refreshed whenever a member is added to the collection.

The acceptable values for this parameter are:

Both	
ConstantUpdate	The collection is refreshed whenever a member is

	added to the collection.
Manual	The collection is refreshed using the Configuration Manager console or the Configuration Manager SDK.
Periodic	The collection is refreshed based on the schedule specified by the RefreshSchedule parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Create a device collection

This command creates a collection for all computers that run Windows 7. The *LimitingCollectionName* parameter specifies that any device in the All Systems collection can be a member of the Windows 7 collection.

```
PS C:\> New-CMDeviceCollection -Name "Windows 7" -LimitingCollectionName "All Systems"
```

Related topics

[Export-CMDeviceCollection](#)

[Get-CMDeviceCollection](#)

[Import-CMDeviceCollection](#)

[Remove-CMDeviceCollection](#)

[Set-CMDeviceCollection](#)

New-CMDeviceCollectionVariable

New-CMDeviceCollectionVariable

Creates a task sequence variable for a device collection in the Configuration Manager hierarchy.

Syntax

Parameter Set: NewByNameMandatory

```
New-CMDeviceCollectionVariable -CollectionName <String> -VariableName <String> [-IsMask <Boolean> ] [-VariableValue <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: NewByIdMandatory

```
New-CMDeviceCollectionVariable -CollectionId <String> -VariableName <String> [-IsMask <Boolean> ] [-VariableValue <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: NewByValueMandatory

```
New-CMDeviceCollectionVariable -Collection <IResultObject> -VariableName <String> [-IsMask <Boolean> ] [-VariableValue <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **New-CMDeviceCollectionVariable** cmdlet creates a task sequence variable for the device collection. You can specify the device collection by its name, ID, or specify on object that represents the collection.

Task sequence variables are a set of name and value pairs that provide a mechanism to configure and customize the steps of a task sequence when the task sequence is deployed to a specific collection. For more information about task sequence variables, see [Planning a Task Sequence Strategy in Configuration Manager](http://go.microsoft.com/fwlink/p/?LinkID=260806) (<http://go.microsoft.com/fwlink/p/?LinkID=260806>) on TechNet.

Parameters

-Collection<IResultObject>

Specifies a Configuration Manager user collection object. To obtain a user collection object, use the **Get-CMUserCollection** cmdlet.

Aliases	none
Required?	true

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CollectionId<String>

Specifies the ID of the device collection.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CollectionName<String>

Specifies the name of the device collection.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-IsMask<Boolean>

Indicates whether a value displays in the Configuration Manager console.

Aliases	none
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-VariableName<String>

Specifies the name of the task sequence variable.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-VariableValue<String>

Specifies a value that you assign to a collection variable in Configuration Manager.

Custom task sequence variables include sets of name and value pairs that supply configuration and operating system deployment settings for a device, operating system and user state configuration tasks on a Configuration Manager client computer.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Create a task sequence variable

This command creates a task sequence variable named New_ComputerName for the device collection named All Unknown Devices.

```
PS C:\> New-CMDeviceCollectionVariable -CollectionName "All Unknown Devices" -VariableName "New_ComputerName"
```

Related topics

[Get-CMDeviceCollectionVariable](#)

[Set-CMDeviceCollectionVariable](#)

[Remove-CMDeviceCollectionVariable](#)

[Get-CMUserCollection](#)



New-CMDeviceVariable

New-CMDeviceVariable

Creates a device variable.

Syntax

Parameter Set: NewByIdMandatory

```
New-CMDeviceVariable -DeviceId <String> -VariableName <String> [-IsMask <Boolean> ] [-VariableValue <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: NewByNameMandatory

```
New-CMDeviceVariable -DeviceName <String> -IsMask <Boolean> -VariableName <String> -VariableValue <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: NewByValueMandatory

```
New-CMDeviceVariable -Device <IResultObject> -VariableName <String> [-IsMask <Boolean> ] [-VariableValue <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **New-CMDeviceVariable** cmdlet creates a device variable for a device. Task sequence processing uses device variables.

Parameters

-Device<IResultObject>

Specifies a **CMDevice** object. To obtain a **CMDevice** object, use the **Get-CMDevice** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeviceId<String>

Specifies a device ID.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeviceName<String>

Specifies a device name. You can specify a NetBIOS name or a fully qualified domain name (FQDN).

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-IsMask<Boolean>

Indicates whether a value displays in the Configuration Manager console.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-VariableName<String>

Specifies the name of the device variable.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-VariableValue<String>

Specifies the value of the variable.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

[Get-CMDeviceVariable](#)

[Remove-CMDeviceVariable](#)

[Set-CMDeviceVariable](#)

[Get-CMDevice](#)

New-CMDistributionPointGroup

New-CMDistributionPointGroup

Creates a distribution point group.

Syntax

Parameter Set: New

```
New-CMDistributionPointGroup -Name <String> [-Description <String> ] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Detailed Description

The **New-CMDistributionPointGroup** cmdlet creates a distribution point group. Distribution point groups provide a logical grouping of distribution points for content distribution.

Parameters

-Description<String>

Specifies a description for the distribution point group.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies a name of the distribution point group.

Aliases	none
---------	------

Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Create a distribution point group

This command creates a distribution point group named DpgDept01 and adds a description for the distribution point group.

```
PS C:\> New-CMDistributionPointGroup -Name "DpgDept01" -Description "Western region"
```

Related topics

[Get-CMDistributionPointGroup](#)

[Set-CMDistributionPointGroup](#)

[Remove-CMDistributionPointGroup](#)

New-CMDriverPackage

New-CMDriverPackage

Creates a driver package.

Syntax

Parameter Set: New

```
New-CMDriverPackage -Name <String> -PackageSourceType {StorageCompress | StorageDirect | StorageLocal | StorageNeedsSpecifying | StorageNOSource} -Path <String> [-Description <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **New-CMDriverPackage** cmdlet creates a driver package.

Parameters

-Description<String>

Specifies a description of a driver package.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies a name for a driver package.

Aliases	none
---------	------

Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PackageSourceType<PackageSourceTypes>

Specifies the method of reading the package source files. Valid values are:

- StorageCompress
- StorageDirect
- StorageLocal
- StorageNeedsSpecifying
- StorageNOSource

The acceptable values for this parameter are:

StorageCompress	
StorageDirect	
StorageLocal	
StorageNeedsSpecifying	
StorageNOSource	

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Path<String>

Specifies a file path to the location where Configuration Manager stores the compressed version of the source files on the site server.

Aliases	PackageSourcePath
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Create a new driver package

This command creates a new driver package by name.

```
PS C:\> New-CMDriverPackage -Name "pckg01" -Path  
"\\Contoso01\Users\pattifuller\Desktop\pckg" -PackageSourceType StorageDirect
```

Related topics

[Export-CMDriverPackage](#)

[Get-CMDriverPackage](#)

[Import-CMDriverPackage](#)

[Remove-CMDriverPackage](#)

[Set-CMDriverPackage](#)

New-CMExchangeClientAccessServer

New-CMExchangeClientAccessServer

Creates a Client Access server role for an Exchange Server.

Syntax

Parameter Set: NewExchangeClientAccessServer

```
New-CMExchangeClientAccessServer -ActiveDirectorySiteName <String> -  
ExchangeClientAccessServerName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **New-CMExchangeClientAccessServer** cmdlet creates a Client Access server role for a Microsoft Exchange Server. The Client Access server role accepts connections to Exchange Server from different types of clients. Software clients such as Microsoft Outlook use POP3 or IMAP4 connections to communicate with Exchange Server. Hardware clients, such as mobile devices, use ActiveSync, POP3, or IMAP4 to communicate with Exchange Server.

Parameters

-ActiveDirectorySiteName<String>

Specifies the name of the ActiveDirectory site on which you are installing the Client Access server role.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ExchangeClientAccessServerName<String>

Specifies the name of the Exchange Client Access server that you create.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Create an Exchange Client Access server

This command creates a new Exchange Client Access server named ContosoWestCAS11 and associates it with the ActiveDirectory site named ContosoWestAD01, then places the resulting Exchange Client Access server object in the variable \$ecs.

```
PS C:\> $ecs= New-CMExchangeClientAccessServer -ExchangeClientAccessServerName  
"ContosoWestCAS11" -ActiveDirectorySiteName "ContosoWestAD01"
```

New-CMExchangeServer

New-CMExchangeServer

Configures a new Exchange Server connector.

Syntax

Parameter Set: New

```
New-CMExchangeServer -DeltaSyncInterval <Int32> -FullSyncSchedule <IResultObject> -
ServerAddress <String> -SiteCode <String> [-ActiveDirectoryContainer <String[]> ] [-
AllowExternalDeviceManagement <Boolean> ] [-ApplicationSetting <Dictionary<String>> ] [-
EmailManagementSetting <Dictionary<String>> ] [-ExchangeClientAccessServer
<Dictionary[]<String>> ] [-GeneralSetting <Dictionary<String>> ] [-MaximumInactiveDay
<Int32> ] [-PasswordSetting <Dictionary<String>> ] [-SecuritySetting <Dictionary<String>> ]
[-UserName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: NewWithIsHostedES

```
New-CMExchangeServer -DeltaSyncInterval <Int32> -FullSyncSchedule <IResultObject> -IsHosted
-ServerAddress <String> -SiteCode <String> [-ActiveDirectoryContainer <String[]> ] [-
AllowExternalDeviceManagement <Boolean> ] [-ApplicationSetting <Dictionary<String>> ] [-
EmailManagementSetting <Dictionary<String>> ] [-GeneralSetting <Dictionary<String>> ] [-
MaximumInactiveDay <Int32> ] [-PasswordSetting <Dictionary<String>> ] [-SecuritySetting
<Dictionary<String>> ] [-UserName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **New-CMExchangeServer** cmdlet configures a new Microsoft Exchange Server connector in Microsoft System Center 2012 Configuration Manager. An Exchange Server connector synchronizes and manages the device enrolled by the Exchange Server. For more information, see [Configuration Manager 2012 Exchange Connector Implementation in Microsoft IT](http://go.microsoft.com/fwlink/?LinkId=286314) (<http://go.microsoft.com/fwlink/?LinkId=286314>) on TechNet.

Parameters

-ActiveDirectoryContainer<String[]>

Specifies an array of names of Active Directory containers. When this parameter is specified, the Exchange Server connector searches the Active Directory containers for the device.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AllowExternalDeviceManagement<Boolean>

Indicates whether an external device management program can manage the device.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ApplicationSetting<Dictionary<String>>

Specifies application settings. For each dictionary entry in the array, specify the setting name as the key and the configuration as the value. Valid values are: AllowUnsignedApplications, AllowUnsignedInstallationPackages, or Block a specific application.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeltaSyncInterval<Int32>

Specifies the interval, in minutes, at which the Exchange Server connector runs delta discovery.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EmailManagementSetting<Dictionary<String>>

Specifies email management settings. For each dictionary entry in the array, specify the setting name as the key and the configuration as the value.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ExchangeClientAccessServer<Dictionary[]<String>>

Specifies an array of Exchange Client Access servers, as key-value pairs.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-FullSyncSchedule<IResultObject>

Specifies a result object that schedules the full discovery time for an Exchange Server connector.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-GeneralSetting<Dictionary<String>>

Specifies general settings. Valid values are:

- RequireManualSyncWhenRoaming
- RequireStorageCardEncryption
- UnapprovedInROMApplicationList
- DevicePolicyRefreshInterval
- MaxInactivityTimeDeviceLock

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-IsHosted

Indicates that the Exchange Server connector configuration is for a hosted or on-premise Exchange Server.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-MaximumInactiveDay<Int32>

Specifies the interval between times that the Exchange Server connector runs device discovery. The cmdlet checks the last sync time of the devices that Exchange Server manages. If the most recent sync time is older than the current time - MinimumInactiveDay, then the exchange connector does not discover the devices.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PasswordSetting<Dictionary<String>>

Specifies password settings. Valid values are:

- AlphanumericDevicePasswordRequired
- DevicePasswordEnabled
- DevicePasswordExpiration
- DevicePasswordHistory
- MaxDevicePasswordFailedAttempts
- PasswordRecoveryEnabled
- MinDevicePasswordComplexCharacters
- MinDevicePasswordLength
- AlphanumericDevicePasswordRequired
- AllowSimpleDevicePassword

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-SecuritySetting<Dictionary<String>>

Specifies a dictionary of security settings. Valid values are:

- AllowBluetooth
- AllowBrowser
- AllowCamera
- AllowDesktopSync
- AllowInternetSharing
- AllowIrDA
- AllowNonProvisionableDevices
- AllowRemoteDesktop
- AllowStorageCard
- AllowTextMessaging
- AllowWiFi

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ServerAddress<String>

Specifies the address of the Exchange Server for which the cmdlet configures the Exchange Server connector.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteCode<String>

Specifies the site code of the Configuration Manager site where a Exchange Server connector runs.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UserName<String>

Specifies the username that the connector uses to connect to the Exchange Server.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

[Get-CMExchangeServer](#)

[Remove-CMExchangeServer](#)

[Set-CMExchangeServer](#)

[Sync-CMExchangeServer](#)

New-CMExchangeServerConnectorAccessRule

New-CMExchangeServerConnectorAccessRule

Configures access settings for a mobile device that uses a Microsoft Exchange Server connector.

Syntax

Parameter Set: New

```
New-CMExchangeServerConnectorAccessRule -AccessLevel {Allow | Block | Quarantine} -Device {DeviceModel | DeviceType} -RuleName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **New-CMExchangeServerConnectorAccessRule** cmdlet configures access settings for a mobile device that uses a Microsoft Exchange Server connector.

Parameters

-AccessLevel<AccessLevelType>

Specifies the type of access for the mobile device. Valid values are:

- Allow
- Block
- Quarantine

The acceptable values for this parameter are:

Allow	
Block	
Quarantine	

Aliases	none
Required?	true
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Device<DeviceType>

Specifies the device type of the mobile device. Valid values are:

-- DeviceModel

-- DeviceType

The acceptable values for this parameter are:

DeviceModel	
DeviceType	

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RuleName<String>

Specifies a name for the access rule.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Configure email management settings for a mobile device

This command creates an access rule for a device type named AccessRule01 that has the Allow access level.

```
PS C:\> New-CMExchangeServerConnectorAccessRule -RuleName "AccessRule01" -AccessLevel Allow -Device DeviceType
```

Related topics

[New-CMExchangeServerConnectorApplicationSetting](#)

[New-CMExchangeServerConnectorEmailManagementSetting](#)

[New-CMExchangeServerConnectorGeneralSetting](#)

[New-CMExchangeServerConnectorPasswordSetting](#)

[New-CMExchangeServerConnectorSecuritySetting](#)

New- CMExchangeServerConnectorApplicationSetting

New-CMExchangeServerConnectorApplicationSetting

Creates application-related settings for a mobile device that uses a Microsoft Exchange Server connector.

Syntax

Parameter Set: New

```
New-CMExchangeServerConnectorApplicationSetting -UnsignedApplication <Boolean> -  
UnsignedInstall <Boolean> [-BlockedApplication <String[]> ] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Detailed Description

The **New-CMExchangeServerConnectorApplicationSetting** cmdlet creates application-related settings for a mobile device that uses a Microsoft Exchange Server connector.

Parameters

-BlockedApplication<String[]>

Specifies an array of names of applications that the mobile device blocks from running.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UnsignedApplication<Boolean>

Indicates whether applications can run at the normal level on a mobile device.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UnsignedInstall<Boolean>

Indicates whether you can install unsigned applications on a mobile device.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Set application options for an Exchange Server connector

This command sets these application options for an Exchange Server connector:

- Allows the mobile device to install unsigned applications.
- Blocks unsigned applications from running on the mobile device.
- Blocks the two applications named a1 and a2 from running.

```
PS C:\> New-CMExchangeServerConnectorApplicationSetting -UnsignedApplication $False -UnsignedInstall $True -BlockedApplication "a1","a2"
```

Related topics

[New-CMExchangeServerConnectorAccessRule](#)

[New-CMExchangeServerConnectorEmailManagementSetting](#)

[New-CMExchangeServerConnectorGeneralSetting](#)

[New-CMExchangeServerConnectorPasswordSetting](#)

[New-CMExchangeServerConnectorSecuritySetting](#)

New- CMExchangeServerConnectorEmailManagementSetting

New-ExchangeServerConnectorEmailManagementSetting

Creates a set of email management settings for a mobile device that uses an Exchange Server connector.

Syntax

Parameter Set: New

```
New-ExchangeServerConnectorEmailManagementSetting -AllowHtmlEmail <Boolean> -ConsumerEmail <Boolean> -EmailAttachmentPolicy <Boolean> -MaximumCalendarAge {All | OneMonth | SixMonths | ThreeMonths | TwoWeeks} -MaximumEmailAge {All | OneDay | OneMonth | OneWeek | ThreeDays | TwoWeeks} -PushWhenRoaming <Boolean> [-MaximumSizeAttachment <Int32> ] [-MaximumSizeHtmlEmail <Int32> ] [-MaximumSizeTextEmail <Int32> ] [-Confirm] [-WhatIf] [<CommonParameters>]
```

Detailed Description

The **New-ExchangeServerConnectorEmailManagementSetting** cmdlet creates a set of e-mail management settings for a mobile device that uses an Exchange Server connector.

Parameters

-AllowHtmlEmail<Boolean>

Indicates whether the mobile devices use HTML for e-mail messages.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-ConsumerEmail<Boolean>

Indicates whether to allow consumer email through the connector.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EmailAttachmentPolicy<Boolean>

Indicates whether the policy allows email attachments.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-MaximumCalenderAge<MaxCalenderAgeType>

Specifies how long a mobile device keeps calendar data. Valid values are:

- All
- OneMonth
- SixMonths
- ThreeMonths
- TwoWeeks

The acceptable values for this parameter are:

All	
OneMonth	
SixMonths	
ThreeMonths	
TwoWeeks	

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-MaximumEmailAge<MaxEmailAgeType>

Specifies how long a mobile device saves email before the device automatically deletes the mail. Valid values are:

- OneDay
- OneMonth
- OneWeek
- ThreeDays
- TwoWeeks

The acceptable values for this parameter are:

All	
OneDay	
OneMonth	
OneWeek	
ThreeDays	
TwoWeeks	

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-MaximumSizeAttachment<Int32>

Specifies the maximum size, in kilobytes (KB), for email attachments.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-MaximumSizeHtmlEmail<Int32>

Specifies the maximum size, in kilobytes, for HTML-formatted email.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-MaximumSizeTextEmail<Int32>

Specifies the maximum size, in kilobytes, for text-formatted email.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PushWhenRoaming<Boolean>

Indicates whether to push email to roaming clients.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Add email management settings to a mobile device

This command creates the following settings for a mobile device:

- Saves email data for one day before erasing it.
- Saves calendar data for three months before erasing it.
- Allows HTML-formatted email.
- Sets a maximum size of 401 KB for text-formatted email and of 402 KB for HTML-formatted email.
- Sets a maximum attachment size of 24 KB.

```
PS C:\> New-CMExchangeServerConnectorEmailManagementSetting -AllowHtmlEmail $True -
ConsumerEmail $True -EmailAttachmentPolicy $True -MaximumCalenderAge ThreeMonths -
MaximumEmailAge OneDay -PushWhenRoaming $True -MaximumSizeAttachment 24 -
MaximumSizeHtmlEmail 402 -MaximumSizeTextEmail 401
```

Related topics

[New-CMExchangeServerConnectorAccessRule](#)

[New-CMExchangeServerConnectorApplicationSetting](#)

[New-CMExchangeServerConnectorGeneralSetting](#)

[New-CMExchangeServerConnectorPasswordSetting](#)

[New-CMExchangeServerConnectorSecuritySetting](#)

New- CMExchangeServerConnectorGeneralSetting

New-**CMExchangeServerConnectorGeneralSetting**

Adds new settings to a Microsoft Exchange Server connector in Configuration Manager.

Syntax

Parameter Set: New

```
New-CMExchangeServerConnectorGeneralSetting -AllowDesktopSync <Boolean> -AllowInternetShare  
<Boolean> -AllowNonProvision <Boolean> [-RefreshInterval <Int32> ] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Detailed Description

The **New-**CMExchangeServerConnectorGeneralSetting**** cmdlet adds new settings to a Microsoft Exchange Server connector connector in Microsoft System Center 2012 Configuration Manager. An Exchange Server connector in System Center 2012 Configuration Manager manages mobile devices that connect to an on-premise or online Exchange Server by using the Exchange ActiveSync protocol.

Parameters

-AllowDesktopSync<Boolean>

Indicates whether the mobile device can synchronize data with desktop machines.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AllowInternetShare<Boolean>

Indicates whether mobile devices can share internet connections with desktop machines.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AllowNonProvision<Boolean>

Indicates whether the connector allows access by mobile devices that System Center 2012 Configuration Manager does not provision.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RefreshInterval<Int32>

Specifies the time, in days, between synchronization runs.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Add general-purpose settings to an Exchange Server connector

This command sets general-purpose options for an Exchange Server connector. The command allows mobile devices to share internet connections and to synchronize with desktop machines. The command allows mobile devices that System Center 2012 Configuration Manager does not provision to use the connector, and sets the synchronization to run every four days.

```
PS C:\> New-CMExchangeServerConnectorGeneralSetting -AllowInternetShare $True - AllowDesktopSync $True -AllowNonProvision $True -RefreshInterval 4
```

Related topics

[New-CMExchangeServerConnectorAccessRule](#)

[New-CMExchangeServerConnectorApplicationSetting](#)

[New-CMExchangeServerConnectorEmailManagementSetting](#)

[New-CMExchangeServerConnectorPasswordSetting](#)

[New-CMExchangeServerConnectorSecuritySetting](#)

New- CMExchangeServerConnectorPasswordSetting

New-**CMExchangeServerConnectorPasswordSetting**

Adds new password settings to a Microsoft Exchange Server connector in Configuration Manager.

Syntax

Parameter Set: New

```
New-CMExchangeServerConnectorPasswordSetting -PasswordEnabled <Boolean> [-  
AllowSimplePassword <Boolean> ] [-MaximumIdleTimeMinutes <Int32> ] [-MinimumComplexChar  
<Int32> ] [-MinimumPasswordLength <Int32> ] [-PasswordComplexity {Pin | Strong} ] [-  
PasswordExpiration <Int32> ] [-PasswordHistory <Int32> ] [-PasswordRecovery <Boolean> ] [-  
WipeAfterFailedAttempt <Int32> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **New-**CMExchangeServerConnectorPasswordSetting**** cmdlet adds new password settings to a Microsoft Exchange Server connector in Microsoft System Center 2012 Configuration Manager. An Exchange Server connector in System Center 2012 Configuration Manager manages mobile devices that connect to an on-premise or online Exchange Server by using the Exchange ActiveSync protocol.

Parameters

-AllowSimplePassword<Boolean>

Indicates whether the mobile device can use simple numeric passwords. A simple numeric password is one that has the same offset between each pair of digits. For example, 2468 is a simple password because each pair of digits has an offset of two. Simple numeric passwords can begin or end with 0 but cannot include values that wrap around in the middle of the digit string, such as 6802.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-MaximumIdleTimeMinutes<Int32>

Specifies the idle time, in minutes, that must elapse before the mobile device locks itself.

The *PasswordEnabled* parameter must also have a value of \$True for the *MaximumIdleTimeMinutes* parameter to take effect.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-MinimumComplexChar<Int32>

Specifies the minimum number of required complex characters in a device password. A complex character is any character that is not a letter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-MinimumPasswordLength<Int32>

Specifies the minimum number of digits, characters, or both for the password.

The *PasswordEnabled* parameter must also have a value of \$True for the *MinimumPasswordLength* parameter to take effect. If *PasswordEnabled* has a value of \$True, you can specify a minimum password length of between 1 and 40 characters. If *PasswordEnabled* is \$False, default password lengths apply: four characters for a simple password (*AllowSimplePassword* set to \$True) or seven characters for an alphanumeric password (*PasswordComplexityType* set to \$True).

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PasswordComplexity<PasswordComplexityType>

Specifies the complexity type for the password. Valid values are:

- Pin: the password must be numeric.
- Strong: the password must be alphanumeric.

The *PasswordEnabled* parameter must also have a value of \$True for the *PasswordComplexity* parameter to take effect.

The acceptable values for this parameter are:

Pin	
Strong	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PasswordEnabled<Boolean>

Indicates whether a user must set a password on the mobile device.

If this parameter is \$True, the user must set a password.

If this parameter is \$False, users can disable their password by using Control Panel and do not need to lock their Windows Mobile device. However, the device does not inform the user that the password is disabled.

If you do not set a value for the parameter, the password-related settings on the mobile device remain in effect.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PasswordExpiration<Int32>

Specifies the number of days before the password expires and the user must enter a new one.

The *PasswordEnabled* parameter must also have a value of \$True for the *PasswordExpiration* parameter to take effect. If *PasswordExpiration* is set to \$False, the user can keep the same password indefinitely. If you do not set a value for *PasswordExpiration*, password expiration settings on the mobile device remain in effect.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PasswordHistory<Int32>

Specifies the number of password changes a user must make before the user can reuse a previous password.

The *PasswordEnabled* parameter must also have a value of \$True for the *PasswordHistory* parameter to take effect. *PasswordExpiration* is set to \$False, users can reuse any previous password. If you do not set a value for *PasswordExpiration*, password settings on the mobile device remain in effect.

Aliases	none
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PasswordRecovery<Boolean>

Indicates whether users can recover a missing device password or PIN from the mobile device. If you do not set a value for this parameter, the password recovery options on the mobile device remain in effect.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-WipeAfterFailedAttempt<Int32>

Specifies the number of failed attempts to reset a password before the device wipes data from itself.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Specify password settings for an Exchange Server connector

This command sets these password-related options for an Exchange Server connector:

- Requires the user to set a password on the mobile device.
- Requires the password to have at least eight characters or digits.
- Causes the password to expire after 51 days.
- Requires 21 password changes before the user can reuse an earlier password.
- Wipes data from the mobile device after six failed attempts to change the password.
- Allows 41 minutes to elapse before the mobile device locks itself.
- Requires an alphanumeric password.
- Allows passwords to be simple.
- Allows users to recover missing passwords from the mobile device.

```
PS C:\> New-CMExchangeServerConnectorPasswordSetting -PasswordEnabled $True -
MinimumPasswordLength 8 -PasswordExpiration 51 -PasswordHistory 21 -WipeAfterFailedAttempt 6
-MaximumIdleTimeMinutes 41 -PasswordComplexity Strong -MinimumComplexChar 3 -
AllowSimplePassword $True -PasswordRecovery $True
```

Related topics

[New-CMExchangeServerConnectorEmailManagementSetting](#)

[New-CMExchangeServerConnectorGeneralSetting](#)

[New-CMExchangeServerConnectorSecuritySetting](#)

New- CMExchangeServerConnectorSecuritySetting

New-CMExchangeServerConnectorSecuritySetting

Configures security options for a Microsoft Exchange Server connector in Configuration Manager.

Syntax

Parameter Set: New

```
New-CMExchangeServerConnectorSecuritySetting -Bluetooth <Boolean> -Camera <Boolean> -  
FileEncrypt <Boolean> -Infra <Boolean> -RemoteDesktop <Boolean> -StorageCard <Boolean> -  
StorageCardEncrypt <Boolean> -TextMessage <Boolean> -WiFiConnection {Allow | Disable |  
HandsfreeOnly} [-Browser <Boolean> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **New-CMExchangeServerConnectorSecuritySetting** cmdlet configures security options for a Microsoft Exchange Server connector in Microsoft System Center 2012 Configuration Manager. An Exchange Server connector in System Center 2012 Configuration Manager manages mobile devices that connect to an on-premise or online Exchange Server by using the Exchange ActiveSync protocol.

Parameters

-Bluetooth<Boolean>

Indicates whether users can run Bluetooth on the mobile device.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Browser<Boolean>

Indicates whether users can use the browser on the mobile device.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Camera<Boolean>

Indicates whether users can use the camera on the mobile device.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-FileEncrypt<Boolean>

Indicates whether users can encrypt files on the mobile device.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Infra<Boolean>

Indicates whether users can use Infrared (IrDA) communications on the device.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RemoteDesktop<Boolean>

Indicates whether a device can initiate a Remote Desktop connection. This policy setting requires an Exchange Enterprise Client Access License.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-StorageCard<Boolean>

Indicates whether the mobile device can access information on a storage card.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-StorageCardEncrypt<Boolean>

Indicates whether the mobile device encrypts new files on the storage card by using a key that is tied to the device.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-TextMessage<Boolean>

Indicates whether the user can send and receive SMS and MMS text messages with the mobile device.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-WiFiConnection<WiFiConnectionType>

Specifies whether the user can use Wireless (Wi-Fi) local area networks (LANs) with the device. Valid values are:

- Allow
- Disable
- HandsfreeOnly

The acceptable values for this parameter are:

Allow	
Disable	
HandsfreeOnly	

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Configure security settings for a mobile device

This command sets the following security options for a mobile device:

- Enables the camera.
- Disables Bluetooth, infrared communications, file encryption on storage cards, and text messaging.
- Allows the mobile device to connect to the Internet only when the device is in handsfree mode.

```
PS C:\> New-CMExchangeServerConnectorSecuritySetting -RemoteDesktop $True -StorageCard $True  
-Camera $True -Bluetooth $False -WiFiConnection HandsfreeOnly -Infra $False -Browser $False  
-StorageCardEncrypt $False -FileEncrypt $False -TextMessage $False
```

Related topics

[New-CMExchangeServerConnectorAccessRule](#)

[New-CMExchangeServerConnectorApplicationSetting](#)

[New-CMExchangeServerConnectorEmailManagementSetting](#)

[New-CMExchangeServerConnectorGeneralSetting](#)

[New-CMExchangeServerConnectorPasswordSetting](#)

New-CMFileReplicationRoute

New-CMFileReplicationRoute

Creates a file replication route for Configuration Manager.

Syntax

Parameter Set: NewFileReplicationRoute

```
New-CMFileReplicationRoute -DestinationSiteCode <String> -SourceSiteCode <String> [-DestinationSiteServerName <String> ] [-FileReplicationAccountName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **New-CMFileReplicationRoute** cmdlet creates a file replication route for Microsoft System Center 2012 Configuration Manager. System Center 2012 Configuration Manager uses file replication routes to transfer file-based data between sites in a hierarchy. Each file replication route identifies a destination site to which file-based data can transfer.

File replication routes were known as addresses in versions of Configuration Manager before System Center 2012 Configuration Manager. The functionality of file replication routes is the same as that of addresses in earlier versions.

Parameters

-DestinationSiteCode<String>

Specifies a destination site for data transfers by using a site code.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DestinationSiteServerName<String>

Specifies a destination site server for data transfers by using a site server name.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-FileReplicationAccountName<String>

Specifies the account that Configuration Manager uses to install a site on the specified server and maintain communications between the site and other sites. This account must have local administrative credentials.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SourceSiteCode<String>

Specifies a source site for data transfers by using a site code.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Create a file replication route

This command creates a file replication route from the site that has the site code IM1 to the site that has the site code IM5 on the server named ImgDataServer01. System Center 2012 Configuration Manager uses the account named AdminRepl01 to manage data transfer over this route.

```
PS C:\> New-CMFileReplicationRoute -DestinationSiteCode "IM5" -SourceSiteCode "IM1" -  
DestinationSiteServerName "ImgDataServer01" -FileReplicationAccountName "AdminRepl01"
```

Related topics

[Get-CMFileReplicationRoute](#)

[Remove-CMFileReplicationRoute](#)

[Set-CMFileReplicationRoute](#)

New-CMGlobalCondition

New-CMGlobalCondition

Creates a global condition in Configuration Manager.

Syntax

Parameter Set: NewADQuery

```
New-CMGlobalCondition -DataType {Boolean | DateTime | FloatingPoint | Integer | IntegerArray | String | StringArray | Version} -DeviceType {Nokia | Windows | WindowsMobile} -DistinguishedName <String> -LdapFilter <String> -Name <String> -Property <String> -SearchScope {Base | OneLevel | Subtree} [-Description <String> ] [-LdapPrefix <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: NewAssembly

```
New-CMGlobalCondition -AssemblyName <String> -DeviceType {Nokia | Windows | WindowsMobile} -Name <String> [-Description <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: NewFileSystem

```
New-CMGlobalCondition -DeviceType {Nokia | Windows | WindowsMobile} -FileOrFolderName <String> -Name <String> -Path <String> [-Description <String> ] [-IncludeSubfolders] [-Is64Bit] [-IsFolder] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: NewFileSystemFile

```
New-CMGlobalCondition -DeviceType {Nokia | Windows | WindowsMobile} -FilePath <String> -Name <String> [-Description <String> ] [-IncludeSubfolders] [-Is64Bit] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: NewIisMetabase

```
New-CMGlobalCondition -DataType {Boolean | DateTime | FloatingPoint | Integer | IntegerArray | String | StringArray | Version} -DeviceType {Nokia | Windows | WindowsMobile} -Name <String> -PropertyId <String> [-Description <String> ] [-MetabasePath <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: NewOmaUri

```
New-CMGlobalCondition -DataType {Boolean | DateTime | FloatingPoint | Integer | IntegerArray | String | StringArray | Version} -DeviceType {Nokia | Windows | WindowsMobile} -Name <String> -OmaUri <String> [-Description <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: NewRegistryKey

```
New-CMGlobalCondition -DeviceType {Nokia | Windows | WindowsMobile} -KeyName <String> -Name <String> -RegistryHive {ClassesRoot | CurrentConfig | CurrentUser | LocalMachine | Users} [-Description <String> ] [-Is64Bit] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: NewRegistryValue

```
New-CMGlobalCondition -DataType {Boolean | DateTime | FloatingPoint | Integer | IntegerArray
```

```
| String | StringArray | Version} -DeviceType {Nokia | Windows | WindowsMobile} -KeyName <String> -Name <String> -RegistryHive {ClassesRoot | CurrentConfig | CurrentUser | LocalMachine | Users} -ValueName <String> [-Description <String> ] [-Is64Bit] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: NewScript

```
New-CMGlobalCondition -DataType {Boolean | DateTime | FloatingPoint | Integer | IntegerArray | String | StringArray | Version} -DeviceType {Nokia | Windows | WindowsMobile} -FilePath <String> -Name <String> -ScriptLanguage {PowerShell | VBScript | JScript | ShellScript} [-Description <String> ] [-Use32BitHost] [-UseLoggedOnUserCredentials] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: NewSqlQueryAllInstances

```
New-CMGlobalCondition -Column <String> -Database <String> -DataType {Boolean | DateTime | FloatingPoint | Integer | IntegerArray | String | StringArray | Version} -DeviceType {Nokia | Windows | WindowsMobile} -FilePath <String> -Name <String> -UseAllInstances [-Description <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: NewSqlQueryDefaultInstance

```
New-CMGlobalCondition -Column <String> -Database <String> -DataType {Boolean | DateTime | FloatingPoint | Integer | IntegerArray | String | StringArray | Version} -DeviceType {Nokia | Windows | WindowsMobile} -FilePath <String> -Name <String> -UseDefaultInstance [-Description <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: NewSqlQuerySpecificInstance

```
New-CMGlobalCondition -Column <String> -Database <String> -DataType {Boolean | DateTime | FloatingPoint | Integer | IntegerArray | String | StringArray | Version} -DeviceType {Nokia | Windows | WindowsMobile} -FilePath <String> -InstanceName <String> -Name <String> [-Description <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: NewWqlQuery

```
New-CMGlobalCondition -Class <String> -DataType {Boolean | DateTime | FloatingPoint | Integer | IntegerArray | String | StringArray | Version} -DeviceType {Nokia | Windows | WindowsMobile} -Name <String> -Property <String> -WhereClause <String> [-Description <String> ] [-Namespace <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: NewXPathQuery

```
New-CMGlobalCondition -DataType {Boolean | DateTime | FloatingPoint | Integer | IntegerArray | String | StringArray | Version} -DeviceType {Nokia | Windows | WindowsMobile} -FilePath <String> -Name <String> -XmlFilePath <String> -XmlNamespace <String[]> [-Description <String> ] [-IncludeSubfolders] [-Is64Bit] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **New-CMGlobalCondition** cmdlet creates a global condition in Microsoft System Center 2012 Configuration Manager. A global condition is a setting or expression in System Center 2012 Configuration Manager that you can use to specify how System Center 2012 Configuration Manager provides and deploys an application to clients.

Parameters

-AssemblyName<String>

Specifies the name of an assembly for which to search. An assembly name must be registered in the Global Assembly Cache.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Class<String>

Specifies a Windows Management Instrumentation (WMI) class used to build a WMI Query Language (WQL) query. The query assesses compliance on client computers.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Column<String>

Specifies the column name used to assess the compliance of the global condition.

Aliases	none
Required?	true
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Database<String>

Specifies the name of a database. The SQL query runs on this database.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DataType<GlobalConditionDataType>

Specifies the global condition data type. Valid values are: Boolean, DateTime, FloatingPoint, Integer, IntegerArray, String, StringArray, and Version.

The acceptable values for this parameter are:

Boolean	
DateTime	
FloatingPoint	
Integer	
IntegerArray	
String	
StringArray	
Version	

Aliases	none
Required?	true
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Description<String>

Specifies a description for the global condition.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeviceType<GlobalConditionDeviceType>

Specifies the type of device to which this global condition applies. Valid values are: Nokia, Windows, and WindowsMobile.

The acceptable values for this parameter are:

Nokia	
Windows	
WindowsMobile	

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DistinguishedName<String>

Specifies the distinguished name of the Active Directory Domain Services (AD DS) object to assess for compliance on client computers.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-FileOrFolderName<String>

Specifies the name of a file or folder. Specify the *IsFolder* parameter to search for a folder.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-FilePath<String>

Specifies a file path for the file that the condition assesses for compliance.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-IncludeSubfolders

Indicates that the global condition searches in subfolders.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InstanceName<String>

Specifies the name of a database instance that the global condition searches. To search the default instance, specify the *UseDefaultInstance* parameter. To search all instances, specify the *UseAllInstances* parameter.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Is64Bit

Indicates that the global condition searches the 64-bit system file location in addition to the 32-bit system file location.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-IsFolder

Indicates that the global condition searches for a folder. If you do not select this parameter, the condition searches for a file. Specify the name of the file or folder by using the *FileOrFolderName* parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-KeyName<String>

Specifies the registry key name for which to search. Use the format *key\subkey*.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-LdapFilter<String>

Specifies an LDAP filter to refine the results from the AD DS query to assess compliance on client computers.

Aliases	none
Required?	true
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-LdapPrefix<String>

Specifies a valid Lightweight Directory Access Protocol (LDAP) prefix for the AD DS query that assesses compliance on client computers. This prefix can be either LDAP:// or GC://.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-MetabasePath<String>

Specifies the path to the metabase file for Internet Information Services (IIS).

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies the name of an IIS metabase file.

Aliases	none
Required?	true
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Namespace<String>

Specifies a namespace from a WMI repository. The default value is Root\cimv2.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OmaUri<String>

Specifies a Uniform Resource Indicator (URI) that points to device-specific parameters for an Open Mobile Alliance (OMA) device.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Path<String>

Specifies the path for an OMA URI.

Aliases	none
Required?	true

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Property<String>

Specifies the property of the AD DS object used to assess compliance on client computers.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PropertyId<String>

Specifies the property of AD DS that Configuration Manager uses to determine client compliance.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RegistryHive<RegistryRootKey>

Specifies the root key in the registry that identifies the registry hive that you search. WMI uses the registry hive to return, set, and change the values of registry keys. Valid values are: ClassesRoot, CurrentConfig, CurrentUser, LocalMachine, and Users.

The acceptable values for this parameter are:

ClassesRoot	
CurrentConfig	
CurrentUser	
LocalMachine	
Users	

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ScriptLanguage<ScriptingLanguage>

Specifies a scripting language to use. Valid values are: PowerShell, VBScript, and JScript.

The acceptable values for this parameter are:

PowerShell	
VBScript	
JScript	
ShellScript	

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SearchScope<SearchScope>

Specifies the search scope in AD DS. Valid values are: Base, OneLevel, and Subtree.

The acceptable values for this parameter are:

Base	
OneLevel	
Subtree	

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Use32BitHost

Indicates that the file or folder is associated with a 64-bit application.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UseAllInstances

Indicates that the global condition searches all database instances. To search a named instance, specify the *InstanceName* parameter. To search the default instance, specify the *UseDefaultInstance* parameter.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UseDefaultInstance

Indicates that the global condition searches the default database instance. To search a named instance, specify the *InstanceName* parameter. To search all instances, specify the *UseAllInstances* parameter.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UseLoggedOnUserCredentials

Indicates that the script runs on client computers by using the logged on user credentials.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ValueName<String>

Specifies the value to be contained in the specified registry key.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-WhereClause<String>

Specifies a WQL query WHERE clause to apply to the specified namespace, class, and property on client computers.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-XmlFilePath<String>

Specifies a file that contains the XML query to use to assess compliance on client computers.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-XmlNamespace<String[]>

Specifies an array of valid, full XML path language (XPath) queries to use to assess compliance on client computers.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Create a global condition

This command creates a global condition that searches the assembly named Microsoft.Office.Tools.Word.v9.0 on Windows devices.

```
PS C:\> New-CMGlobalCondition -AssemblyName "Microsoft.Office.Tools.Word.v9.0" -DeviceType $Windows
```

Related topics

[Get-CMGlobalCondition](#)

[Remove-CMGlobalCondition](#)

[Set-CMGlobalCondition](#)

New-CMHardwareRequirement

New-CMHardwareRequirement

Creates a Configuration Manager hardware requirement object for a product.

Syntax

Parameter Set: New

```
New-CMHardwareRequirement -MinCpu <Int32> -MinDiskFree <Int64> -MinDiskSize <Int64> -MinRam <Int64> -Product <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **New-CMHardwareRequirement** cmdlet creates a hardware requirement object for a product.

Microsoft System Center 2012 Configuration Manager manages Asset Intelligence information, including hardware requirements, for different software products. You can add, modify, or delete your own hardware requirements, but you cannot change built-in hardware requirement objects.

Parameters

-MinCpu<Int32>

Specifies the minimum CPU speed, in megahertz (MHz), required for a software product.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-MinDiskFree<Int64>

Specifies the minimum amount of available disk memory, in kilobytes (KB), required for a software product.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-MinDiskSize<Int64>

Specifies the minimum disk size, in kilobytes, required for a software product.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-MinRam<Int64>

Specifies the minimum amount of random access memory (RAM), in kilobytes, required for a software product.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Product<String>

Specifies of the name of a software product.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Create a hardware requirement object

This command creates a hardware requirement object for a software product called Accounts Program. The command specifies the minimum hardware requirements for the product. If you do not include all of these required parameters, the system prompts you for values.

```
PS C:\> New-CMHardwareRequirement -MinCpu 233 -MinDiskFree 1572864 -MinDiskSize 10485760 -MinRam 131072 -Product "Accounts Program"
```

```
IsLocal      :  
MinCPU       : 233  
MinDiskFree  : 1572864  
MinDiskSize  : 10485760  
MinRAM       : 131072  
Product      : Accounts Program  
State        :
```

Related topics

[Get-CMHardwareRequirement](#)

[Remove-CMHardwareRequirement](#)

[Set-CMHardwareRequirement](#)

New-CMInstallationSourceFile

New-CMInstallationSourceFile

Creates an installation source file for Configuration Manager.

Syntax

Parameter Set: NewInstallationSourceFilesByNetworkLocation

```
New-CMInstallationSourceFile -CopyFromNetworkLocation -UncPath <String> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: NewInstallationSourceFilesByParent

```
New-CMInstallationSourceFile -CopyFromParentSiteServer [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: NewInstallationSourceFilesBySecondaryLocation

```
New-CMInstallationSourceFile -CopyFromSecondarySiteLocation -LocalPath <String> [-Confirm] [  
-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **New-CMInstallationSourceFile** cmdlet creates an installation source file for Microsoft System Center 2012 Configuration Manager. An installation source file is an object that contains installation source parameters for a secondary site installation. A secondary site has no System Center 2012 Configuration Manager database of its own. Instead, it forwards information that it gets from clients to a primary site that stores the data for all the secondary sites that are attached to it.

Parameters

-CopyFromNetworkLocation

Indicates that Configuration Manager copies the installation files for a secondary site installation from a specific Universal Naming Convention (UNC) path.

Aliases	none
Required?	true
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CopyFromParentSiteServer

Indicates that Configuration Manager copies the installation files for a secondary site installation from the primary site.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CopyFromSecondarySiteLocation

Indicates that the installation files for a secondary site installation reside on the secondary site server.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-LocalPath<String>

Specifies a path to source files in the local file system of the secondary site server.

Aliases	none
Required?	true
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UncPath<String>

Specifies a UNC path to source files in the local file system of the secondary site server.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Create an installation source file

This command creates an installation source file for a secondary site installation by copying the installation files from the primary site.

```
PS C:\> New-CMInstallationSourceFile -CopyFromParentSiteServer
```

New-CMMaintenanceWindow

New-CMMaintenanceWindow

Creates a maintenance window for a collection.

Syntax

Parameter Set: ByScheduleMandatory

```
New-CMMaintenanceWindow [-CollectionID] <String> -Name <String> -Schedule <IResultObject> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SoftwareUpdateOnly

```
New-CMMaintenanceWindow [-CollectionID] <String> -ApplyToSoftwareUpdateOnly -Name <String> -Schedule <IResultObject> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: TaskSequenceOnly

```
New-CMMaintenanceWindow [-CollectionID] <String> -ApplyToTaskSequenceOnly -Name <String> -Schedule <IResultObject> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **New-CMMaintenanceWindow** cmdlet creates a maintenance window for a collection.

Maintenance windows are periods of time reserved for write operations such as applying software updates, installing software, or configuring computer settings.

Parameters

-ApplyToSoftwareUpdateOnly

Indicates that the maintenance window is used to apply software updates only.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

-ApplyToTaskSequenceOnly

Indicates that the maintenance window is used to apply task sequences only.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

-CollectionID<String>

Specifies the ID of the collection that the maintenance window applies to.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies the name of the maintenance window.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

-Schedule<IResultObject>

Specifies a **CMSchedule** object. The schedule specifies when the maintenance window occurs. To create a **CMSchedule** object, use the **New-CMSchedule** cmdlet.

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

Example 1: Create a maintenance window

The first command uses the **New-CMSchedule** cmdlet to create a schedule object, and then stores it in the \$MWSchedule variable.

The second command creates a maintenance window named MonthlySchedule for the specified collection. The maintenance window uses the schedule stored in the \$MWSchedule variable.

```
PS C:\> $MWSchedule = New-CMSchedule -DayOfWeek Friday -DurationCount 0 -DurationInterval  
Hours -RecurCount 1 -Start "10/12/2013 21:00:00"
```

```
PS C:\> New-CMMaintenanceWindow -CollectionID "AAA0005D" -Name "MonthlySchedule" -Schedule  
$MWSchedule
```

Related topics

[Get-CMMaintenanceWindow](#)

[Remove-CMMaintenanceWindow](#)

[Set-CMMaintenanceWindow](#)

[New-CMSchedule](#)

New-CMMigrationJob

New-CMMigrationJob

Creates a migration job in System Center 2012 Configuration Manager.

Syntax

Parameter Set: NewMigrationJobByCollectionMigrateObject

```
New-CMMigrationJob -CollectionMigrationJobType -MigrateObjectWithSpecifiedCollection -
MigrationCollection <IResultObject[]> -MigrationObject <IResultObject[]> -Name <String> -
SecurityScope <IResultObject[]> [-CollectionLimitingMapping <Hashtable> ] [-
ContentObjectsSiteCodeMapping <Hashtable> ] [-Description <String> ] [-
EnableProgramAfterAdvertisementMigrated <Boolean> ] [-MigrationJobSchedule <DateTime> ] [-
OverwriteAllObject <Boolean> ] [-SaveCollectionInfoPath <String> ] [-SaveObjectInfoPath
<String> ] [-SiteCodeReplacementMapping <Hashtable> ] [-
TransferOrganizationalFolderStructure <Boolean> ] [-UtcTime <Boolean> ] [-Confirm] [-WhatIf]
[ <CommonParameters>]
```

Parameter Set: NewMigrationJobByCollectionNotMigrateObject

```
New-CMMigrationJob -CollectionMigrationJobType -MigrationCollection <IResultObject[]> -Name
<String> -SecurityScope <IResultObject[]> [-CollectionLimitingMapping <Hashtable> ] [-
Description <String> ] [-EnableProgramAfterAdvertisementMigrated <Boolean> ] [-
MigrationJobSchedule <DateTime> ] [-OverwriteAllObject <Boolean> ] [-SaveCollectionInfoPath
<String> ] [-SaveObjectInfoPath <String> ] [-SiteCodeReplacementMapping <Hashtable> ] [-
TransferOrganizationalFolderStructure <Boolean> ] [-UtcTime <Boolean> ] [-Confirm] [-WhatIf]
[ <CommonParameters>]
```

Parameter Set: NewMigrationJobByObject

```
New-CMMigrationJob -MigrationObject <IResultObject[]> -Name <String> -ObjectMigrationJobType
-SecurityScope <IResultObject[]> [-ContentObjectsSiteCodeMapping <Hashtable> ] [-Description
<String> ] [-MigrationJobSchedule <DateTime> ] [-OverwriteAllObject <Boolean> ] [-
SaveObjectInfoPath <String> ] [-SiteCodeReplacementMapping <Hashtable> ] [-
TransferOrganizationalFolderStructure <Boolean> ] [-UtcTime <Boolean> ] [-Confirm] [-WhatIf]
[ <CommonParameters>]
```

Parameter Set: NewMigrationJobByObjectModified

```
New-CMMigrationJob -MigrationObject <IResultObject[]> -Name <String> -
ObjectModifiedAfterMigrationJobType -SecurityScope <IResultObject[]> [-
ContentObjectsSiteCodeMapping <Hashtable> ] [-Description <String> ] [-MigrationJobSchedule
<DateTime> ] [-OverwriteAllObject <Boolean> ] [-SaveObjectInfoPath <String> ] [-
SiteCodeReplacementMapping <Hashtable> ] [-TransferOrganizationalFolderStructure <Boolean> ]
[-UtcTime <Boolean> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **New-CMMigrationJob** cmdlet creates a migration job in Microsoft System Center 2012 Configuration Manager.

Parameters

-CollectionLimitingMapping<Hashtable>

Specifies key-value pairings to limit a collection. Collection limiting prevents the addition of collection members you do want in the collection.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CollectionMigrationJobType

Indicates that the job migrates collections, objects, or previously migrated objects.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ContentObjectsSiteCodeMapping<Hashtable>

Specifies key-value pairs that map content objects in the new site.

Aliases	none
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Description<String>

Specifies a description for the migration job.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnableProgramAfterAdvertisementMigrated<Boolean>

Indicates whether to enable programs associated with an advertisement after they have migrated.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-MigrateObjectWithSpecifiedCollection

Indicates that you migrate associated objects with the collection.

Aliases	none
Required?	true

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-MigrationCollection<IResultObject[]>

Specifies an array of input objects.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-MigrationJobSchedule<DateTime>

Specifies a date time, in D.HH:MM:SS format, to schedule the migration job.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-MigrationObject<IResultObject[]>

Specifies an array of input objects.

Aliases	none
Required?	true

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies the name of a migration job in Configuration Manager.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ObjectMigrationJobType

Indicates that the job type is an object migration job.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ObjectModifiedAfterMigrationJobType

Indicates that the new migration job only includes objects that were modified since the last migration.

Aliases	none
Required?	true

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OverwriteAllObject<Boolean>

Indicates whether to overwrite objects in the destination database.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SaveCollectionInfoPath<String>

Specifies a path for the collection information.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SaveObjectInfoPath<String>

Specifies a path for the object information.

Aliases	none
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SecurityScope<IResultObject[]>

Specifies an array of security scope objects. To obtain a security scope object, use the **Get-CMSecurityScope** cmdlet. The cmdlet applies the security scopes that you specify to data migrated to the destination hierarchy.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteCodeReplacementMapping<Hashtable>

Specifies key-value pairs that map a migrated collection to a site in the destination.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-TransferOrganizationalFolderStructure<Boolean>

Indicates whether to migrate an empty collection. Configuration Manager converts the empty collection to an organizational folder.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UtcTime<Boolean>

Indicates whether to use UTC time.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Create a migration job

The first command creates mappings of site codes and stores the result in the \$Group variable.

The second command uses the **Get-CMInitModifiableSecuredCategory** cmdlet and stores the result in the \$CategoryObjects variable.

The third command uses the **Get-CMMigrationEntity** cmdlet and stores the result in the \$MigrationEntity variable.

The fourth command uses the **New-CMMigrationJob** cmdlet to create a migration job.

```
PS C:\> $Group = @{"122" = "123" "234" = "123"}
PS C:\> $CategoryObjects = Get-CMInitModifiableSecuredCategory
PS C:\> $MigrationEntity = Get-CMMigrationEntity
PS C:\> New-CMMigrationJob -Name "123" -ObjectMigrationJobType -
ContentObjectsSiteCodeMapping $Group -SecurityScope $CategoryObjects -MigrationObject
$MigrationEntity
```

Related topics

[Get-CMInitialModifiableSecuredCategory](#)

[Get-CMMigrationEntity](#)

[Get-CMSecurityScope](#)

New-CMOperatingSystemImage

New-CMOperatingSystemImage

Creates an operating system image.

Syntax

Parameter Set: New

```
New-CMOperatingSystemImage -Name <String> -Path <String> [-Description <String> ] [-Version <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **New-CMOperatingSystemImage** cmdlet adds an operating system image to a Microsoft System Center 2012 Configuration Manager site. Operating system images are .wim format files and represent a compressed collection of reference files and folders that are System Center 2012 Configuration Manager requires to successfully install and configure an operating system on a computer.

Parameters

-Description<String>

Specifies a description for the operating system image.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies the name of an operating system image.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Path<String>

Specifies the network path to the installation source files of an operating system image source .wim file.

Aliases	PkgSourcePath
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Version<String>

Specifies a version of the operating system image.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Create an operating system image

This command creates the operating system image named STANDARD_WIN7 and specifies the network path to the installation source files of the operating system image.

```
PS C:\> New-CMOperatingSystemImage -Name "STANDARD_WIN7" -Path
"\\Contoso01\CM\Images\STANDARD_WIN7.wim"
```

Related topics

[Get-CMOperatingSystemImage](#)

[Set-CMOperatingSystemImage](#)

[Remove-CMOperatingSystemImage](#)

[Get-CMOperatingSystemImageUpdateSchedule](#)

New-CMOperatingSystemInstaller

New-CMOperatingSystemInstaller

Adds an operating system installer.

Syntax

Parameter Set: New

```
New-CMOperatingSystemInstaller -Name <String> -Path <String> [-Description <String> ] [-Version <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **New-CMOperatingSystemInstaller** cmdlet adds an operating system installer to a Microsoft System Center 2012 Configuration Manager site. An operating system installer is an installation package that contains all the files that Microsoft System Center 2012 Configuration Manager needs to install a Windows operating system on a reference computer.

Parameters

-Description<String>

Specifies a description for an operating system installer.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies a name of the operating system installer.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Path<String>

Specifies the network path to the installation source files of an operating system installer.

Aliases	PkgSourcePath
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Version<String>

Specifies the version of an operating system installer.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Add an operating system installer

This command adds an operating system installer named INSTALL01 and specifies the network path to the installation source files of the operating system installer.

```
PS C:\> New-CMOperatingSystemInstaller -Name "INSTALL01" -Path "\\Contoso01\CM\Win8Install"
```

Related topics

[Get-CMOperatingSystemInstaller](#)

[Remove-CMOperatingSystemInstaller](#)

[Set-CMOperatingSystemInstaller](#)

New-CMPackage

New-CMPackage

Creates a Configuration Manager package.

Syntax

Parameter Set: New

```
New-CMPackage -Name <String> [-Description <String> ] [-Language <String> ] [-Manufacturer <String> ] [-Path <String> ] [-Version <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: NewPackageByDefinitionNoSourceFileWithExisted

```
New-CMPackage -FromDefinition -PackageDefinitionName <String> -PackageNoSourceFile [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: NewPackageByDefinitionNoSourceFileWithNew

```
New-CMPackage -FromDefinition -PackageNoSourceFile -PackagePath <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: NewPackageByDefinitionSourceFileWithExisted

```
New-CMPackage -FromDefinition -PackageDefinitionName <String> -SourceFileType {AlwaysObtainSourceFile | CreateCompressedVersionOfSourceFile} -SourceFolderPath <String> -SourceFolderPathType {LocalFolderOnSiteServer | UncNetworkPath} [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: NewPackageByDefinitionSourceFileWithNew

```
New-CMPackage -FromDefinition -PackagePath <String> -SourceFileType {AlwaysObtainSourceFile | CreateCompressedVersionOfSourceFile} -SourceFolderPath <String> -SourceFolderPathType {LocalFolderOnSiteServer | UncNetworkPath} [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **New-CMPackage** cmdlet creates a Microsoft System Center 2012 Configuration Manager package. A package is a System Center 2012 Configuration Manager object that contains the content files and instructions for distributing programs, software updates, boot images, operating system images, and drivers to System Center 2012 Configuration Manager clients.

Parameters

-Description<String>

Specifies a description for the package. You can use a maximum of 128 characters.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-FromDefinition

Indicates that Configuration Manager creates the package from a package definition file.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Language<String>

Specifies the language version of the package. You can use a maximum of 32 characters in a format that you choose to use to identify the language version. Configuration Manager uses the Language property together with Manufacturer, Name, and Version to identify a package. For example, you can have an English version and a German version of the same package.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Manufacturer<String>

Specifies a manufacturer name to help you identify the package. You can use a maximum of 32 characters.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies a name for the package.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PackageDefinitionName<String>

Specifies the name of a package definition file.

Aliases	none
Required?	true
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PackageNoSourceFile

Indicates that the package does not require source files to be present on client devices.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PackagePath<String>

Specifies a share name or path that Configuration Manager creates for the package source files on distribution points.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Path<String>

Specifies the location of the files to add to the package.

You can specify either a full local path or a UNC path. Make sure that this location contains all the files and subdirectories that the program needs to complete, including any scripts.

Aliases	PackageSourcePath
---------	-------------------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SourceFileType<SourceFileType>

Specifies the source file type. Valid values are:

- AlwaysObtainSourceFile
- CreateCompressedVersionOfSourceFile

The acceptable values for this parameter are:

AlwaysObtainSourceFile	
CreateCompressedVersionOfSourceFile	

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SourceFolderPath<String>

Specifies the location of the source files for the package.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-SourceFolderPathType<SourceFolderPathType>

Specifies the source folder path type. Valid values are:

-- LocalFolderOnSiteServer

-- UncNetworkPath

The acceptable values for this parameter are:

LocalFolderOnSiteServer	
UncNetworkPath	

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Version<String>

Specifies a version number for the package.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Create a package

This command creates a Configuration Manager package named ScriptsPackage01.

```
PS C:\> New-CMPackage -Name "ScriptsPackage01"
```

Example 2: Create a package and add a description

This command creates a Configuration Manager package named ScriptsPackage02 and adds the specified description to the package.

```
PS C:\> New-CMPackage -Name "ScriptsPackage02" -Description "This package deploys scripts  
that run on a recurring schedule."
```

Related topics

[Export-CMPackage](#)

[Get-CMPackage](#)

[Import-CMPackage](#)

[Remove-CMPackage](#)

[Set-CMPackage](#)

New-CMProgram

New-CMProgram

Creates a new program in Configuration Manager.

Syntax

Parameter Set: NewStandardProgram

```
New-CMProgram -CommandLine <String> -PackageName <String> -StandardProgramName <String> [-DiskSpaceRequirement <String> ] [-DiskSpaceUnit {GB | KB | MB} ] [-DriveLetter <String> ] [-DriveMode {RenameWithUnc | RequiresDriveLetter | RequiresSpecificDriveLetter} ] [-Duration <Int32> ] [-ProgramRunType {OnlyWhenNoUserIsLoggedOn | OnlyWhenUserIsLoggedOn | WhetherOrNotUserIsLoggedOn} ] [-Reconnect <Boolean> ] [-RunMode {RunWithAdministrativeRights | RunWithUserRights} ] [-RunType {Hidden | Maximized | Minimized | Normal} ] [-UserInteraction <Boolean> ] [-WorkingDirectory <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: NewDeviceProgram

```
New-CMProgram -CommandLine <String> -DeviceProgramName <String> -PackageName <String> [-CommandLineFolder <String> ] [-Comment <String> ] [-DiskSpaceRequirement <String> ] [-DiskSpaceUnit {GB | KB | MB} ] [-DownloadProgramType {AsSoonAsPossible | OnlyOverFastNetwork | OnlyWhenTheDeviceIsDocked} ] [-Requirement <String> ] [-WorkingDirectory <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **New-CMProgram** cmdlet creates a new program in Microsoft System Center 2012 Configuration Manager. Programs are commands that are associated with a System Center 2012 Configuration Manager package. Programs identify the actions that occur when the client receives the client package. You can associate multiple programs with the same package.

Parameters

-CommandLine<String>

Specifies the command line for the program.

Aliases	none
Required?	true

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CommandLineFolder<String>

Specifies the folder that contains the executable program. This folder can be an absolute path on the client, or a path relative to the distribution folder that contains the package.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Comment<String>

Specifies optional text about a program, such as a description. On client computers, this text is displayed in Run Advertised Programs in Control Panel.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeviceProgramName<String>

Specifies a device program name.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DiskSpaceRequirement<String>

Specifies the amount of disk space that the software program requires to run on the computer. If a value is specified, units for the value must also be specified. The value must be greater than or equal to zero.

Aliases	DiskSpaceReq
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DiskSpaceUnit<DiskSpaceUnitType>

Specifies the units, GB, KB, or MB, for the *DiskSpaceRequirement* parameter.

The acceptable values for this parameter are:

GB	
KB	
MB	

Aliases	none
Required?	false
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DownloadProgramType<DownloadProgramType>

Specifies when the program is to run. Valid values are: AsSoonAsPossible, OnlyOverFastNetwork, and OnlyWhenTheDevicesLocked.

The acceptable values for this parameter are:

AsSoonAsPossible	
OnlyOverFastNetwork	
OnlyWhenTheDevicesDocked	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DriveLetter<String>

Specifies a drive letter to qualify the location if the *DriveMode* parameter is used.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DriveMode<DriveModeType>

Indicates whether the program requires a specific drive letter, specified in the *DriveLetter* parameter. By default, the program runs with a Universal Naming Convention (UNC) name. If *DriveMode* is set to *RequiresDriveLetter*, the program uses any available drive letter. If *DriveMode* is set to *RequiresSpecificDriveLetter*, the program only runs if the drive is not already in use.

The acceptable values for this parameter are:

RenameWithUnc	
RequiresDriveLetter	
RequiresSpecificDriveLetter	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Duration<Int32>

Specifies the maximum amount of time the program is expected to run. The default value is 120 minutes.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PackageName<String>

Specifies a package name.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ProgramRunType<ProgramRunType>

Specifies the logon conditions that are necessary for the program to run. Valid values are: OnlyWhenNoUserIsLoggedIn, OnlyWhenUserIsLoggedIn, and WhetherOrNotUserIsLoggedIn. The default setting is OnlyWhenUserIsLoggedIn.

The acceptable values for this parameter are:

OnlyWhenNoUserIsLoggedIn	
OnlyWhenUserIsLoggedIn	
WhetherOrNotUserIsLoggedIn	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Reconnect<Boolean>

Indicates whether the client computer reconnects to the distribution point when the user logs on.

Aliases	none
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Requirement<String>

Specifies additional requirements for standard or device programs.

Aliases	Requirements
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RunMode<RunModeType>

Specifies the credentials that the program requires to run on the client computer, either RunWithAdministrativeRights or RunWithUserRights.

The acceptable values for this parameter are:

RunWithAdministrativeRights	
RunWithUserRights	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RunType<RunType>

Specifies the mode in which the program will run on the client computer. Valid values are: Hidden, Maximized, Minimized, and Normal. The default is Normal.

The acceptable values for this parameter are:

Hidden	
Maximized	
Minimized	
Normal	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-StandardProgramName<String>

Specifies the standard program name.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UserInteraction<Boolean>

Indicates whether to allow users to interact with the program.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-WorkingDirectory<String>

Specifies a working directory for the program.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Create a new standard program

This command creates a new standard program in Configuration Manager.

```
PS C:\> New-CMProgram -PackageName "test" -StandardProgramName SPM -CommandLine "RunMe" -WorkingDirectory "C:\temp" -RunType Hidden -ProgramRunType OnlyWhenNoUserIsLoggedIn -DiskSpaceRequirement 100 -DiskSpaceUnit GB -Duration 100 -DriveMode RenameWithUnc
```

Example 2: Create a new device program

This command creates a new device program in Configuration Manager.

```
PS C:\> New-CMProgram -PackageName "Contoso-12" -DeviceProgramName DPM -Comment "Upgrades for December" -WorkingDirectory "C:\temp" -CommandLine "RunMe" -CommandLineFolder "C:\Windows\" -DiskSpaceRequirement 10 -DiskSpaceUnit GB -DownloadProgramType OnlyWhenTheDeviceIsDocked -Requirement "All previous updates"
```

Related topics

[Disable-CMProgram](#)

[Enable-CMProgram](#)

[Get-CMProgram](#)

[Remove-CMProgram](#)

[Set-CMProgram](#)

New- CMRemoteConnectionProfileConfigurationItem

New-**CMRemoteConnectionProfileConfigurationItem**

Creates a remote connection profile.

Syntax

Parameter Set: New

```
New-CMRemoteConnectionProfileConfigurationItem -Name <String> [-Description <String> ] [-  
EnableNLA <Boolean> ] [-EnablePrimaryUsers <Boolean> ] [-EnableTSConnection <Boolean> ] [-  
EnableTSFirewallRule <Boolean> ] [-RDGatewayServer <String> ] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Detailed Description

The **New-**CMRemoteConnectionProfileConfigurationItem**** cmdlet creates a remote connection profile. Client computers use remote connection profiles to remotely connect to computers from outside the domain or over the Internet.

Parameters

-Description<String>

Specifies a description for a remote connection profile.

Aliases	LocalizedDescription
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnableNLA<Boolean>

Indicates whether to allow connections only from computers that run Remote Desktop by using Network Level Authentication.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnablePrimaryUsers<Boolean>

Indicates whether all primary users of the client computer are allowed to remotely connect. If you specify a value for this parameter, you must specify values for the *EnableTSConnection* and *EnableTSFirewallRule* parameters.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnableTSConnection<Boolean>

Indicates whether to allow remote connections to client computers. If you specify a value for this parameter, you must specify values for the *EnablePrimaryUsers* and *EnableTSFirewallRule* parameters.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnableTSFirewallRule<Boolean>

Indicates whether to allow Windows Firewall exceptions for connections in Windows domains and on private networks. If you specify a value for this parameter, you must specify values for the *EnablePrimaryUsers* and *EnableTSConnections* parameters.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies a name for a remote connection profile.

Aliases	LocalizedDisplayName
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RDGatewayServer<String>

Specifies the host name and port of the Remote Desktop gateway server, for example, Boston.gateway.Contoso.com:8080.

Aliases	none
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

[Copy-CMRemoteConnectionProfileConfigurationItem](#)

[Get-CMRemoteConnectionProfileConfigurationItem](#)

[Remove-CMRemoteConnectionProfileConfigurationItem](#)

[Set-CMRemoteConnectionProfileConfigurationItem](#)

New-CMSchedule

New-CMSchedule

Creates a schedule token.

Syntax

Parameter Set: RecurrenceNone

```
New-CMSchedule -Nonrecurring [-IsUtc] [-ScheduleString] [-Start <DateTime> ] [ <CommonParameters>]
```

Parameter Set: RecurMonthlyByWeekday

```
New-CMSchedule -DayOfWeek {Sunday | Monday | Tuesday | Wednesday | Thursday | Friday | Saturday} -WeekOrder <ScheduleWeekOrder> [-IsUtc] [-RecurCount <Int32> ] [-ScheduleString] [-Start <DateTime> ] [ <CommonParameters>]
```

Parameter Set: RecurMonthlyByWeekdayWithDuration

```
New-CMSchedule -DayOfWeek {Sunday | Monday | Tuesday | Wednesday | Thursday | Friday | Saturday} -DurationCount <Int32> -DurationInterval <ScheduleInterval> -WeekOrder <ScheduleWeekOrder> [-IsUtc] [-RecurCount <Int32> ] [-ScheduleString] [-Start <DateTime> ] [ <CommonParameters>]
```

Parameter Set: RecurMonthlyByWeekdayWithEnd

```
New-CMSchedule -DayOfWeek {Sunday | Monday | Tuesday | Wednesday | Thursday | Friday | Saturday} -End <DateTime> -WeekOrder <ScheduleWeekOrder> [-IsUtc] [-RecurCount <Int32> ] [-ScheduleString] [-Start <DateTime> ] [ <CommonParameters>]
```

Parameter Set: RecurMonthlyLastDayOfMonth

```
New-CMSchedule -LastDayOfMonth [-IsUtc] [-RecurCount <Int32> ] [-ScheduleString] [-Start <DateTime> ] [ <CommonParameters>]
```

Parameter Set: RecurMonthlyLastDayOfMonthWithDuration

```
New-CMSchedule -DurationCount <Int32> -DurationInterval <ScheduleInterval> -LastDayOfMonth [-IsUtc] [-RecurCount <Int32> ] [-ScheduleString] [-Start <DateTime> ] [ <CommonParameters>]
```

Parameter Set: RecurMonthlyLastDayOfMonthWithEnd

```
New-CMSchedule -End <DateTime> -LastDayOfMonth [-IsUtc] [-RecurCount <Int32> ] [-ScheduleString] [-Start <DateTime> ] [ <CommonParameters>]
```

Parameter Set: RecurrenceInterval

```
New-CMSchedule -RecurCount <Int32> -RecurInterval <ScheduleInterval> [-IsUtc] [-ScheduleString] [-Start <DateTime> ] [ <CommonParameters>]
```

Parameter Set: RecurrenceIntervalWithDuration

```
New-CMSchedule -DurationCount <Int32> -DurationInterval <ScheduleInterval> -RecurCount <Int32> -RecurInterval <ScheduleInterval> [-IsUtc] [-ScheduleString] [-Start <DateTime> ] [ <CommonParameters>]
```

Parameter Set: RecurrenceIntervalWithEnd

New-CMSchedule -End <DateTime> -RecurCount <Int32> -RecurInterval <ScheduleInterval> [-IsUtc] [-ScheduleString] [-Start <DateTime>] [<CommonParameters>]

Parameter Set: RecurrenceMonthlyByDate

New-CMSchedule -DayOfMonth <Int32> [-IsUtc] [-RecurCount <Int32>] [-ScheduleString] [-Start <DateTime>] [<CommonParameters>]

Parameter Set: RecurrenceMonthlyByDateWithDuration

New-CMSchedule -DayOfMonth <Int32> -DurationCount <Int32> -DurationInterval <ScheduleInterval> [-IsUtc] [-RecurCount <Int32>] [-ScheduleString] [-Start <DateTime>] [<CommonParameters>]

Parameter Set: RecurrenceMonthlyByDateWithEnd

New-CMSchedule -DayOfMonth <Int32> -End <DateTime> [-IsUtc] [-RecurCount <Int32>] [-ScheduleString] [-Start <DateTime>] [<CommonParameters>]

Parameter Set: RecurrenceNoneWithDuration

New-CMSchedule -DurationCount <Int32> -DurationInterval <ScheduleInterval> -Nonrecurring [-IsUtc] [-ScheduleString] [-Start <DateTime>] [<CommonParameters>]

Parameter Set: RecurrenceNoneWithEnd

New-CMSchedule -End <DateTime> -Nonrecurring [-IsUtc] [-ScheduleString] [-Start <DateTime>] [<CommonParameters>]

Parameter Set: RecurrenceWeekly

New-CMSchedule -DayOfWeek {Sunday | Monday | Tuesday | Wednesday | Thursday | Friday | Saturday} [-IsUtc] [-RecurCount <Int32>] [-ScheduleString] [-Start <DateTime>] [<CommonParameters>]

Parameter Set: RecurrenceWeeklyWithDuration

New-CMSchedule -DayOfWeek {Sunday | Monday | Tuesday | Wednesday | Thursday | Friday | Saturday} -DurationCount <Int32> -DurationInterval <ScheduleInterval> [-IsUtc] [-RecurCount <Int32>] [-ScheduleString] [-Start <DateTime>] [<CommonParameters>]

Parameter Set: RecurrenceWeeklyWithEnd

New-CMSchedule -DayOfWeek {Sunday | Monday | Tuesday | Wednesday | Thursday | Friday | Saturday} -End <DateTime> [-IsUtc] [-RecurCount <Int32>] [-ScheduleString] [-Start <DateTime>] [<CommonParameters>]

Detailed Description

The **New-CMSchedule** cmdlet creates a schedule token. In Microsoft System Center 2012 Configuration Manager, you use schedule tokens to configure scheduling information. You can create schedule tokens to schedule events with differing frequencies such as daily, weekly, and monthly.

Use the **Convert-CMSchedule** cmdlet to decode and encode schedule tokens into and from an interval string. You can then use the interval strings to set schedule properties when you define or modify System Center 2012 Configuration Manager objects.

Parameters

-DayOfMonth<Int32>

Specifies the day of the month when the event occurs. Valid values range from 0 through 31. The default value is 0, which indicates the last day of the month.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DayOfWeek<DayOfWeek>

Specifies the day of the week when the event occurs. Valid values are:

- Sunday (default)
- Monday
- Tuesday
- Wednesday
- Thursday
- Friday
- Saturday

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?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DurationCount<Int32>

Specifies the number of days during which the scheduled event occurs. Valid values range from 0 through 31. The default value is 0, which indicates that the scheduled action continues indefinitely.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DurationInterval<ScheduleInterval>

Specifies the time when the event occurs.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-End<DateTime>

Specifies the date and time when the scheduled event ends.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-IsUtc

Indicates that the time is Coordinated Universal Time (UTC).

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-LastDayOfMonth

Indicates that the event occurs monthly on the last day of the month.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Nonrecurring

Indicates that the scheduled event does not recur.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RecurCount<Int32>

Specifies the number of recurrences of the scheduled event.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RecurInterval<ScheduleInterval>

Specifies the time when the scheduled event recurs.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ScheduleString

Indicates that the schedule token is converted to an interval string.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Start<DateTime>

Specifies the date and time when the scheduled event occurs.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-WeekOrder<ScheduleWeekOrder>

Specifies the week of the month when the event occurs. Valid values are:

- 0. Last (default)
- 1. First
- 2. Second
- 3. Third
- 4. Fourth

Aliases	none
Required?	true
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

<CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about_CommonParameters](#).

Examples

Example 1: Create a schedule token

This command creates a schedule token that specifies that the event occurs the last day of the month at the specified date and time.

```
PS C:\> New-CMSchedule -DayOfMonth 0 -DateTime "20120105185728.303000+000"
```

Related topics

[Convert-CMSchedule](#)

New-CMSecondarySite

New-CMSecondarySite

Creates a secondary site in Configuration Manager.

Syntax

Parameter Set: NewDistributionPointByHTTPAndCreateCertificate

```
New-CMSecondarySite -CertificateExpirationTimeUtc <DateTime> -CreateSelfSignedCertificate -  
Http -InstallationSourceFile <IResultObject[]> -InstallInternetServer <Boolean> -  
ParentSiteCode <String> -ServerName <String> -SiteCode <String> -SiteName <String> -  
SqlServerSetting <IResultObject[]> [-AllowFallbackForContent <Boolean> ] [-AllowPreStaging  
<Boolean> ] [-BoundaryGroups <IResultObject[]> ] [-ContentMonitoringPriority {High | Highest  
| Low | Lowest | Medium} ] [-EnableAnonymous <Boolean> ] [-InstallationFolder <String> ] [-  
MinFreeSpaceMB <Int32> ] [-PrimaryContentLibraryLocation {A | Automatic | B | C | D | E | F  
| G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z} ] [-  
PrimaryPackageShareLocation {A | Automatic | B | C | D | E | F | G | H | I | J | K | L | M |  
N | O | P | Q | R | S | T | U | V | W | X | Y | Z} ] [-SecondaryContentLibraryLocation {A |  
Automatic | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U |  
V | W | X | Y | Z} ] [-SecondaryPackageShareLocation {A | Automatic | B | C | D | E | F | G  
| H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z} ] [-  
ValidateContentSchedule <IResultObject> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: NewDistributionPointByHTTPAndImportCertificate

```
New-CMSecondarySite -CertificatePassword <SecureString> -CertificatePath <String> -Http -  
ImportCertificate -InstallationSourceFile <IResultObject[]> -InstallInternetServer <Boolean>  
-ParentSiteCode <String> -ServerName <String> -SiteCode <String> -SiteName <String> -  
SqlServerSetting <IResultObject[]> [-AllowFallbackForContent <Boolean> ] [-AllowPreStaging  
<Boolean> ] [-BoundaryGroups <IResultObject[]> ] [-ContentMonitoringPriority {High | Highest  
| Low | Lowest | Medium} ] [-EnableAnonymous <Boolean> ] [-ForceWhenDuplicateCertificate  
<Boolean> ] [-InstallationFolder <String> ] [-MinFreeSpaceMB <Int32> ] [-  
PrimaryContentLibraryLocation {A | Automatic | B | C | D | E | F | G | H | I | J | K | L | M  
| N | O | P | Q | R | S | T | U | V | W | X | Y | Z} ] [-PrimaryPackageShareLocation {A |  
Automatic | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U |  
V | W | X | Y | Z} ] [-SecondaryContentLibraryLocation {A | Automatic | B | C | D | E | F |  
G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z} ] [-  
SecondaryPackageShareLocation {A | Automatic | B | C | D | E | F | G | H | I | J | K | L | M  
| N | O | P | Q | R | S | T | U | V | W | X | Y | Z} ] [-ValidateContentSchedule  
<IResultObject> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: NewDistributionPointByHTTPSAndCreateCertificate

```
New-CMSecondarySite -CertificateExpirationTimeUtc <DateTime> -CreateSelfSignedCertificate -  
Https -InstallationSourceFile <IResultObject[]> -InstallInternetServer <Boolean> -  
ParentSiteCode <String> -ServerName <String> -SiteCode <String> -SiteName <String> -  
SqlServerSetting <IResultObject[]> [-AllowFallbackForContent <Boolean> ] [-AllowPreStaging  
<Boolean> ] [-BoundaryGroups <IResultObject[]> ] [-ClientConnectionType {Internet |  
InternetAndIntranet | Intranet} ] [-ContentMonitoringPriority {High | Highest | Low | Lowest
```

```

| Medium} ] [-InstallationFolder <String> ] [-MinFreeSpaceMB <Int32> ] [-
PrimaryContentLibraryLocation {A | Automatic | B | C | D | E | F | G | H | I | J | K | L | M
| N | O | P | Q | R | S | T | U | V | W | X | Y | Z} ] [-PrimaryPackageShareLocation {A |
Automatic | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U |
V | W | X | Y | Z} ] [-SecondaryContentLibraryLocation {A | Automatic | B | C | D | E | F |
G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z} ] [-
SecondaryPackageShareLocation {A | Automatic | B | C | D | E | F | G | H | I | J | K | L | M
| N | O | P | Q | R | S | T | U | V | W | X | Y | Z} ] [-ValidateContentSchedule
<IResultObject> ] [-Confirm] [-WhatIf] [ <CommonParameters>]

```

Parameter Set: NewDistributionPointByHTTPSAndImportCertificate

```

New-CMSecondarySite -CertificatePassword <SecureString> -CertificatePath <String> -Https -
ImportCertificate -InstallationSourceFile <IResultObject[]> -InstallInternetServer <Boolean>
-ParentSiteCode <String> -ServerName <String> -SiteCode <String> -SiteName <String> -
SqlServerSetting <IResultObject[]> [-AllowFallbackForContent <Boolean> ] [-AllowPreStaging
<Boolean> ] [-BoundaryGroups <IResultObject[]> ] [-ClientConnectionType {Internet |
InternetAndIntranet | Intranet} ] [-ContentMonitoringPriority {High | Highest | Low | Lowest
| Medium} ] [-ForceWhenDuplicateCertificate <Boolean> ] [-InstallationFolder <String> ] [-
MinFreeSpaceMB <Int32> ] [-PrimaryContentLibraryLocation {A | Automatic | B | C | D | E | F
| G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z} ] [-
PrimaryPackageShareLocation {A | Automatic | B | C | D | E | F | G | H | I | J | K | L | M |
N | O | P | Q | R | S | T | U | V | W | X | Y | Z} ] [-SecondaryContentLibraryLocation {A |
Automatic | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U |
V | W | X | Y | Z} ] [-SecondaryPackageShareLocation {A | Automatic | B | C | D | E | F | G
| H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z} ] [-
ValidateContentSchedule <IResultObject> ] [-Confirm] [-WhatIf] [ <CommonParameters>]

```

Detailed Description

The **New-CMSecondarySite** cmdlet creates a secondary site in Microsoft System Center 2012 Configuration Manager.

Parameters

-AllowFallbackForContent<Boolean>

Indicates whether clients can use a fallback source location for content.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-AllowPreStaging<Boolean>

Indicates whether the secondary site can pre-stage contents.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-BoundaryGroups<IResultObject[]>

Specifies an array of boundary group objects. To get a boundary group object, use the [Get-CMBoundaryGroup](#) cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CertificateExpirationTimeUtc<DateTime>

Specifies the date and time at which the self-signed certificate expires for a distribution point on this secondary site.

Aliases	none
Required?	true
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CertificatePassword<SecureString>

Specifies the password for the PKI imported certificate for the distribution point.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CertificatePath<String>

Specifies the import path for the PKI issued certificate that the distribution point uses. This parameter applies when the secondary site has installed and configured IIS to create a distribution point.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ClientConnectionType<ClientConnectionTypes>

Specifies a client connection type. Valid values are:

- Internet
- InternetAndIntranet
- Intranet

The acceptable values for this parameter are:

Internet	
InternetAndIntranet	
Intranet	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ContentMonitoringPriority<Priority>

Specifies content monitoring priority. Valid values are:

- High
- Highest
- Low
- Lowest
- Medium

The acceptable values for this parameter are:

High	
Highest	
Low	
Lowest	
Medium	

Aliases	none
Required?	false
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CreateSelfSignedCertificate

Indicates that the secondary site creates a self-signed certificate.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnableAnonymous<Boolean>

Indicates whether client computers communicate anonymously with the distribution point. This parameter applies when the secondary site has installed and configured IIS to create a distribution point.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ForceWhenDuplicateCertificate<Boolean>

Indicates whether Configuration Manager overwrites a duplicate certificate when you import a PKI client certificate for the secondary site.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Http

Indicates that client computers communicate with the distribution point by using HTTP. This parameter applies when the secondary site has installed and configured IIS to create a distribution point. This option does not support mobile devices or Mac computers.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Https

Indicates that client computers communicate with the distribution point by using HTTPS. This parameter applies when the secondary site has installed and configured IIS to create a distribution point. This option does not support mobile devices or Mac computers.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ImportCertificate

Indicates that the cmdlet imports a PKI certificate instead of using a self-signed certificate for the distribution point.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InstallationFolder<String>

Specifies the installation folder on the secondary site server where the cmdlet installs the site files.

Aliases	InstallDir
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InstallationSourceFile<IResultObject[]>

Specifies an array of installation source file objects for Configuration Manager. To obtain an installation source file object, use the **New-CMInstallationSourceFile** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InstallInternetServer<Boolean>

Specifies whether to install and configure IIS if Configuration Manager requires it. This parameter must be \$True before the cmdlet can install the site system role for the distribution point on this secondary site.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-MinFreeSpaceMB<Int32>

Specifies the amount of space, in megabytes, to reserve on each drive that the distribution point uses. This value determines the remaining free space on the drive after the distribution stores content on the drive.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ParentSiteCode<String>

Specifies the site code of the primary site that is parent to the secondary site that the cmdlet creates.

Aliases	none
Required?	true
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PrimaryContentLibraryLocation<DriveType>

Specifies a primary content library location. Valid values are:

- Automatic.
- Drive letter from A: through Z:.

The acceptable values for this parameter are:

A	
Automatic	
B	
C	
D	
E	
F	
G	
H	
I	
J	
K	
L	
M	
N	
O	
P	
Q	
R	
S	
T	

U	
V	
W	
X	
Y	
Z	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PrimaryPackageShareLocation<DriveType>

Specifies a primary package share location. Valid values are:

- Automatic.
- Drive letter from A: through Z:.

The acceptable values for this parameter are:

A	
Automatic	
B	
C	
D	
E	
F	
G	
H	
I	

J	
K	
L	
M	
N	
O	
P	
Q	
R	
S	
T	
U	
V	
W	
X	
Y	
Z	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SecondaryContentLibraryLocation<DriveType>

Specifies a secondary content library location. Valid values are:

- Automatic.
- Drive letter from A: through Z:.

The acceptable values for this parameter are:

A	
Automatic	
B	
C	
D	
E	
F	
G	
H	
I	
J	
K	
L	
M	
N	
O	
P	
Q	
R	
S	
T	
U	
V	
W	
X	
Y	
Z	

Aliases	none
Required?	false
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SecondaryPackageShareLocation<DriveType>

Specifies a secondary package share location. Valid values are:

- Automatic.
- Drive letter from A: through Z:.

The acceptable values for this parameter are:

A	
Automatic	
B	
C	
D	
E	
F	
G	
H	
I	
J	
K	
L	
M	
N	
O	
P	
Q	
R	
S	
T	

U	
V	
W	
X	
Y	
Z	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ServerName<String>

Specifies an FQDN for the secondary site server.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteCode<String>

Specifies the site code that uniquely identifies the secondary site in the hierarchy.

Aliases	none
Required?	true

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteName<String>

Specifies the site name that helps identify the secondary site.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SqlServerSetting<IResultObject[]>

Specifies an array of SQL Server settings object in Configuration Manager. To obtain a SQL Server settings object, use the **New-CMSqlServerSetting** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ValidateContentSchedule<IResultObject>

Specifies an object that represents a schedule type and determines how frequently System Center 2012 Configuration Manager validates the integrity of packages on the distribution point.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Create a secondary site

This first command creates a SQL Server settings object and specifies that Microsoft SQL Server Express is copied to a Configuration Manager secondary site. The command stores the object in the \$CIObj variable.

The second command creates a secondary site named Contoso remote site that has the site code CM2 on the server named server2.corp.contoso.com. The command specifies the SQL Server settings object for the secondary site stored in \$CIObj. The command specifies the primary site for the secondary site that has the site code CM1.

```
PS C:\> $CIObj = New-CMSqlServerSetting -CopySqlServerExpressOnSecondarySite
PS C:\> New-CMSecondarySite -CertificateExpirationTimeUtc "2/1/2020 12:00 AM" -
CreateSelfSignedCertificate -Https -InstallationSourceFile "\\ContosoServer1\SourceFiles" -
InstallInternetServer $True -ParentSiteCode "CM1" -ServerName "server2.corp.contoso.com" -
SiteCode "CM2" -SiteName "Contoso remote site" -SqlServerSetting $CIObj
```

Related topics

[Remove-CMSecondarySite](#)

[New-CMInstallationSourceFile](#)

[New-CMSqlServerSetting](#)

New-CMSecurityScope

New-CMSecurityScope

Creates a security scope.

Syntax

Parameter Set: New

```
New-CMSecurityScope -Name <String> [-Description <String> ] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Detailed Description

The **New-CMSecurityScope** cmdlet creates a security scope. Security scopes in Microsoft System Center 2012 Configuration Manager provide administrative users with access to securable objects. Security scopes are a named set of securable objects that are assigned to administrator users as a group. All securable objects must be assigned to one or more security scopes.

Parameters

-Description<String>

Specifies the description of a security scope.

Aliases	CategoryDescription
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies the name of the security scope.

Aliases	CategoryName
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	true

-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

Example 1: Create a security scope

This command creates a security scope named ScopeT01 and adds a description to the new security scope.

```
PS C:\> New-CMSecurityScope -Name "ScopeT01" -Description "Security scope for team 1."
```

Related topics

[Get-CMSecurityScope](#)

[Set-CMSecurityScope](#)

[Remove-CMSecurityScope](#)

[Remove-CMSecurityScopeFromAdministrativeUser](#)

New-CMSiteSystemServer

New-CMSiteSystemServer

Creates an object that represents a site system server in Configuration Manager.

Syntax

Parameter Set: New

```
New-CMSiteSystemServer -ServerName <String> -SiteCode <String> [-AccountName <String> ] [-EnableProxy <Boolean> ] [-FdmOperation <Boolean> ] [-ProxyAccessAccount <IResultObject> ] [-ProxyServerName <String> ] [-ProxyServerPort <UInt32> ] [-PublicFqdn <String> ] [-UseSiteServerAccount] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **New-CMSiteSystemServer** cmdlet creates an object that represents a site system server in Microsoft System Center 2012 Configuration Manager. A site system server provides functionality to a configuration management site, such as communication between a System Center 2012 Configuration Manager server and System Center 2012 Configuration Manager clients.

You can designate a new server as a site system server and add the site system roles, or install site system roles to an existing site system server in the site.

Parameters

-AccountName<String>

Specifies an account name for the Configuration Manager site.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

-EnableProxy<Boolean>

Indicates whether to enable a proxy server.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

-FdmOperation<Boolean>

Indicates whether a site system pushes information back to a site server, or whether a site server pushes information to a site system.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

-ProxyAccessAccount<IResultObject>

Specifies an input object that contains the domain and user name to authenticate with the proxy server. Do not use User Principal Name (UPN) format.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

-ProxyServerName<String>

Specifies the name of a proxy server. Use a fully qualified domain name (FQDN), short name, or IPv4/IPv6 address.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

-ProxyServerPort<UInt32>

Specifies the port number of a proxy server to use to connect to the Internet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

-PublicFqdn<String>

Specifies a fully qualified domain name (FQDN) path for the Configuration Manager site.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false

Accept Wildcard Characters?	false
-----------------------------	-------

-ServerName<String>

Specifies a server name for the Configuration Manager site.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

-SiteCode<String>

Specifies a site code for the Configuration Manager site.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UseSiteServerAccount

Indicates that the cmdlet uses a user account name for the site.

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

Example 1: Create a new site system server

This command creates a new site system server with the server name and site parameters.

```
PS C:\> New-CMSiteSystemServer -ServerName "ContosoSvr02" -SiteCode "CM2"
```

Related topics

[Get-CMSiteSystemServer](#)

[Set-CMSiteSystemServer](#)

New-CMSoftwareMeteringRule

New-CMSoftwareMeteringRule

Creates a Configuration Manager software metering rule.

Syntax

Parameter Set: New

```
New-CMSoftwareMeteringRule -Path <String> -SiteCode <String> [-Comment <String> ] [-FileVersion <String> ] [-LanguageId <Int32> ] [-OriginalFileName <String> ] [-ProductName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **New-CMSoftwareMeteringRule** cmdlet creates a Microsoft System Center 2012 Configuration Manager software metering rule. A software metering rule specifies a piece of software, along with version information. You can obtain necessary file information from Windows Explorer.

Software metering monitors and collects software usage data from System Center 2012 Configuration Manager clients, such as when clients began using a particular software program and how long users have worked with that software. You can create software metering rules that specify which software to monitor.

For more information about software metering in System Center 2012 Configuration Manager, see [Introduction to Software Metering in Configuration Manager](http://go.microsoft.com/fwlink/?LinkId=268432) (<http://go.microsoft.com/fwlink/?LinkId=268432>) on TechNet.

Parameters

-Comment<String>

Specifies a comment for a software metering rule.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-FileVersion<String>

Specifies a version of the software that a rule meters.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-LanguageId<Int32>

Specifies a LocaleID of the software that a rule meters. For more information and a list of locale identifiers, see the [Locale IDs Assigned by Microsoft](http://go.microsoft.com/fwlink/?LinkId=262651) topic at <http://go.microsoft.com/fwlink/?LinkId=262651>.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OriginalFileName<String>

Specifies an original file name of the software that a rule meters. This parameter can differ from the *Path* parameter if a user changed the name.

Aliases	none
Required?	false
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Path<String>

Specifies a file path of the software that a rule meters.

Aliases	FileName
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ProductName<String>

Specifies a name for a product that a rule meters.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteCode<String>

Specifies a site code for a Configuration Manager site.

Aliases	none
Required?	true
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Create a software metering rule

This command creates a software metering rule for the System Center 2012 Configuration Manager site CM1. The command specifies the file name, version, original file name, and product name for the software product.

```
PS C:\> New-CMSoftwareMeteringRule -Path "Notepad.exe" -SiteCode "CM1" -FileVersion  
"6.1.7600.16385" -OriginalFileName "NOTEPAD.EXE" -ProductName "Microsoft Windows Operating  
System"
```

```
ApplyToChildSites : True  
Comment           :  
Enabled           : True  
FileName          : Notepad.exe  
FileVersion       : 6.1.7600.16385  
LanguageID       :  
LastUpdateTime    :  
OriginalFileName  : NOTEPAD.EXE  
ProductName       : Microsoft Windows Operating System  
RuleID           :  
SecurityKey       :  
SiteCode         : CM1  
SourceSite       :
```

Related topics

[Disable-CMSoftwareMeteringRule](#)

[Enable-CMSoftwareMeteringRule](#)

[Get-CMSoftwareMeteringRule](#)

[Remove-CMSoftwareMeteringRule](#)

[Set-CMSoftwareMeteringRule](#)

New-CMSoftwareUpdateAutoDeploymentRule

New-CMSoftwareUpdateAutoDeploymentRule

Creates Configuration Manager deployment rules for automatic software updates.

Syntax

Parameter Set: New

```
New-CMSoftwareUpdateAutoDeploymentRule -CollectionName <String> -DeploymentPackageName <String> -Name <String> [-AddToExistingSoftwareUpdateGroup <Boolean> ] [-AlertTime <Int32> ] [-AlertTimeUnit {Days | Hours | Months | Weeks} ] [-AllowRestart <Boolean> ] [-AllowSoftwareInstallationOutsideMaintenanceWindow <Boolean> ] [-AllowUseMeteredNetwork <Boolean> ] [-ArticleId <String[]> ] [-AvailableImmediately <Boolean> ] [-AvailableTime <Int32> ] [-AvailableTimeUnit {Days | Hours | Months | Weeks} ] [-BulletinId <String[]> ] [-CustomSeverity {Critical | Important | Low | Moderate | None} ] [-DateReleasedOrRevised {Last10months | Last11months | Last12hours | Last14days | Last16hours | Last1day | Last1hour | Last1month | Last1year | Last20hours | Last21days | Last28days | Last2days | Last2hours | Last2months | Last3days | Last3hours | Last3months | Last4days | Last4hours | Last4months | Last5days | Last5months | Last6days | Last6months | Last7days | Last7months | Last8hours | Last8months | Last9months} ] [-DeadlineImmediately <Boolean> ] [-DeadlineTime <Int32> ] [-DeadlineTimeUnit {Days | Hours | Months | Weeks} ] [-DeployWithoutLicense <Boolean> ] [-Description <String> ] [-DisableOperationManager <Boolean> ] [-DownloadFromInternet <Boolean> ] [-DownloadFromMicrosoftUpdate <Boolean> ] [-EnabledAfterCreate <Boolean> ] [-GenerateOperationManagerAlert <Boolean> ] [-GenerateSuccessAlert <Boolean> ] [-Language <String[]> ] [-LanguageSelection <String[]> ] [-Location <String> ] [-MicrosoftAsVendor <Boolean> ] [-NoInstallOnRemote <Boolean> ] [-NoInstallOnUnprotected <Boolean> ] [-Product <String[]> ] [-Required <String[]> ] [-RunType {DoNotRunThisRuleAutomatically | RunTheRuleAfterAnySoftwareUpdatePointSynchronization | RunTheRuleOnSchedule} ] [-Schedule <IResultObject> ] [-SendWakeUpPacket <Boolean> ] [-Severity {Critical | Important | Low | Moderate | None} ] [-SuccessPercentage <Int32> ] [-Superseded <Boolean> ] [-SuppressRestartServer <Boolean> ] [-SuppressRestartWorkstation <Boolean> ] [-Title <String[]> ] [-UpdateClassification <String[]> ] [-UpdateDescription <String[]> ] [-UseBranchCache <Boolean> ] [-UserNotification {DisplayAll | DisplaySoftwareCenterOnly | HideAll} ] [-UseUtc <Boolean> ] [-VerboseLevel {AllMessages | OnlyErrorMessages | OnlySuccessAndErrorMessage} ] [-WriteFilterHandling <Boolean> ] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Detailed Description

The **New-CMSoftwareUpdateAutoDeploymentRule** cmdlet creates Microsoft System Center 2012 Configuration Manager deployment rules for automatic software updates.

System Center 2012 Configuration Manager uses rules to manage automatic deployment of software updates. When a rule runs, System Center 2012 Configuration Manager adds updates that qualify for

the rule to a software update group. The System Center 2012 Configuration Manager server downloads content files and copies them to distribution points, and then updates client computers.

Parameters

-AddToExistingSoftwareUpdateGroup<Boolean>

Indicates whether the rule adds to an existing update group. If this value is \$True, each time the rule runs and finds new updates, it adds them to an existing update group. If this value is \$False, it creates a new update group. Specify the existing update group or assign a name for the new update group by using the *DeploymentPackageName* parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AlertTime<Int32>

Specifies an integer offset from an update deployment deadline. The rule uses this value to specify when the rule generates alerts. Specify a time unit by using the *AlertTimeUnit* parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AlertTimeUnit<TimeUnitType>

Specifies a unit of time for the *AlertTime* parameter. Valid values are:

- Days
- Hours

-- Months

-- Weeks

The acceptable values for this parameter are:

Days	
Hours	
Months	
Weeks	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AllowRestart<Boolean>

Indicates whether to allow a computer to restart if the update deployment takes place outside of a maintenance window. A maintenance window is a specified period of time used for computer maintenance and updates. If this value is \$True, Configuration Manager restarts the computer, if necessary, to complete the update. If this value is \$False, Configuration Manager does not restart the computer.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AllowSoftwareInstallationOutsideMaintenanceWindow<Boolean>

Indicates whether the update deployment takes place even if scheduled outside of a maintenance window. A maintenance window is a specified period of time used for computer maintenance and updates. If this value is \$True, Configuration Manager deploys the update even the scheduled time falls outside the service window. If this value is \$False, Configuration Manager does not deploy the update outside the service window, but Configuration Manager waits until it can deploy in a service window.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AllowUseMeteredNetwork<Boolean>

Indicates whether to allow clients to download content over a metered Internet connection after the deadline, which may incur additional expense.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ArticleId<String[]>

Specifies an array of criteria, as strings, for software updates. The rule adds software updates that have article IDs that meet specified criteria to the software update group.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AvailableImmediately<Boolean>

Indicates whether this rule deploys updates as soon as the updates become available. If you select a value of \$False, use the *AvailableTime* and *AvailableTimeUnit* parameters to specify how long after the rule runs to deploy the updates.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AvailableTime<Int32>

Specifies a period of time as an integer. Configuration Manager deploys the updates this long after the rule runs. Specify a time unit by using the *AvailableTimeUnit* parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AvailableTimeUnit<TimeUnitType>

Specifies a unit of time for the *AvailableTime* parameter. Valid values are:

- Days
- Hours
- Months
- Weeks

The acceptable values for this parameter are:

Days	
Hours	
Months	
Weeks	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-BulletinId<String[]>

Specifies an array of criteria, as strings, for software updates. The rule adds software updates that have bulletin IDs that meet specified criteria to the software update group.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CollectionName<String>

Specifies the name of device collection or user collection.

Aliases	none
Required?	true

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CustomSeverity<SeverityType[]>

Specifies an array of custom severity types for software updates. The rule adds software updates that have custom severity levels that meet specified criteria to the software update group. Valid values are:

- Critical
- Important
- Low
- Moderate
- None

The acceptable values for this parameter are:

Critical	
Important	
Low	
Moderate	
None	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DateReleasedOrRevised<DateReleasedOrRevisedType>

Specifies a date released or revised for software updates. The rule adds software updates that have a date that meets specified criteria to the software update group. Valid values are:

-
- Last10months
 - Last11months
 - Last12hours
 - Last14days
 - Last16hours
 - Last1day
 - Last1hour
 - Last1month
 - Last1year
 - Last20hours
 - Last21days
 - Last28days
 - Last2days
 - Last2hours
 - Last2months
 - Last3days
 - Last3hours
 - Last3months
 - Last4days
 - Last4hours
 - Last4months
 - Last5days
 - Last5months
 - Last6days
 - Last6months
 - Last7days
 - Last7months
 - Last8hours
 - Last8months
 - Last9months

The acceptable values for this parameter are:

Last10months	
Last11months	
Last12hours	
Last14days	
Last16hours	

Last1day	
Last1hour	
Last1month	
Last1year	
Last20hours	
Last21days	
Last28days	
Last2days	
Last2hours	
Last2months	
Last3days	
Last3hours	
Last3months	
Last4days	
Last4hours	
Last4months	
Last5days	
Last5months	
Last6days	
Last6months	
Last7days	
Last7months	
Last8hours	
Last8months	
Last9months	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-DeadlineImmediately<Boolean>

Indicates whether to impose the deadline is as soon as the rule runs. If you specify a value of \$False, use the *DeadlineTime* and *DeadlineTimeUnit* parameters to specify how long after the rule runs to set the deadline. After the deadline, Configuration Manager installs required updates.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeadlineTime<Int32>

Specifies a period of time as an integer. The deadline for updates is this long after the rule runs. Specify a time unit by using the *DeadlineTimeUnit* parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeadlineTimeUnit<TimeUnitType>

Specifies a unit of time for the *DeadlineTime* parameter. Valid values are:

- Days
- Hours
- Months
- Weeks

The acceptable values for this parameter are:

Days	
Hours	
Months	
Weeks	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeploymentPackageName<String>

Specifies the name of a software update deployment package.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeployWithoutLicense<Boolean>

Indicates whether the rule deploys updates without licenses. If you specify a value of \$True, Configuration Manager deploys all updates for this rule and approves any license agreements. If this value is \$False, Configuration Manager deploys only updates that do not include a license or for which the license agreement has been approved.

Aliases	none
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Description<String>

Specifies a description for the automatic deployment rule for software updates.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DisableOperationManager<Boolean>

Indicates whether to disable System Center 2012 – Operations Manager alerts during software updates.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DownloadFromInternet<Boolean>

Indicates whether computers download software updates from the Internet. If you specify a value of `$False`, specify an alternative location where computers can download updates by using the *Location* parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DownloadFromMicrosoftUpdate<Boolean>

Indicates whether computers download content from Microsoft Update if that content is unavailable on a preferred distribution point of remote distribution point.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnabledAfterCreate<Boolean>

Indicates whether to enable software deployment for the associated software update group after this rule runs. If this value is \$False, deploy the software update group manually.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-GenerateOperationManagerAlert<Boolean>

Indicates whether to generate Operations Manager alerts during a software update.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-GenerateSuccessAlert<Boolean>

Indicates whether to generate an alert for successful deployment.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Language<String[]>

Specifies an array of criteria, as strings, for software updates. The rule adds software updates that have languages that meet specified criteria to the software update group.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-LanguageSelection<String[]>

Specifies an array of languages, as strings. Computers download software updates available in the specified languages, in addition to non-language-specific updates.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Location<String>

Specifies a location in your network where computers can download software updates. In order to use this location, specify a value of `$False` for the *DownloadFromInternet* parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-MicrosoftAsVendor<Boolean>

Indicates whether the rule includes only updates that have Microsoft as the vendor.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies a name for the automatic deployment rule for software updates.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-NoInstallOnRemote<Boolean>

Indicates whether to disallow installation of updates on remote systems. If you specify a value of \$True, if the client is within a slow or unreliable network boundary, or when the client uses a fallback source location for content, then Configuration Manager does not install software updates. If you specify a value of \$False, installation proceeds.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-NoInstallOnUnprotected<Boolean>

Indicates whether to disallow installation of updates on unprotected systems. If you specify a value of \$True, if software updates are not available on any preferred distribution points, Configuration Manager does not download and install software updates. If you specify a value of \$False, installation proceeds.

Aliases	none
Required?	false
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Product<String[]>

Specifies an array of criteria, as strings, for software updates. The rule adds software updates for products that meet specified criteria to the software update group.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Required<String[]>

Specifies an array of criteria, as strings, for software updates. The rule adds software updates identified by required that meet specified criteria to the software update group.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RunType<RunType>

Specifies the mode in which an update runs on the client computer. Valid values are:

- DoNotRunThisRuleAutomatically
- RunTheRuleAfterAnySoftwareUpdatePointSynchronization
- RunTheRuleOnSchedule

If you specify `RunTheRuleOnSchedule`, specify a schedule by using the *Schedule* parameter.
The acceptable values for this parameter are:

<code>DoNotRunThisRuleAutomatically</code>	
<code>RunTheRuleAfterAnySoftwareUpdatePointSynchronization</code>	
<code>RunTheRuleOnSchedule</code>	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Schedule<IResultObject>

Specifies a schedule object for the deployment. To obtain a schedule object, use the **New-CMSchedule** cmdlet. Specify a schedule for this parameter if you specify a value of `RunTheRuleOnSchedule` for the *RunType* parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SendWakeUpPacket<Boolean>

Indicates whether to send a wake up packet to computers before the deployment begins. If this value is `$True`, Configuration Manager wakes a computer from sleep. If this value is `$False`, it does not wake computers from sleep. For computers to wake, you must first configure Wake On LAN.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Severity<SeverityType[]>

Specifies an array of severity levels for software updates. The rule adds software updates for specified severity types to the software update group. Valid values are:

- Critical
- Important
- Low
- Moderate
- None

The acceptable values for this parameter are:

Critical	
Important	
Low	
Moderate	
None	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SuccessPercentage<Int32>

Specifies a percentage for client compliance as an integer from 0 to 99. If compliance falls below this percentage, Configuration Manager produces optional alerts.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Superseded<Boolean>

Indicates whether the rule adds updates superseded by other updates.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SuppressRestartServer<Boolean>

Indicates whether to suppress a required update for a server. Some software updates require a system restart to complete the installation process.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SuppressRestartWorkstation<Boolean>

Indicates whether to suppress a required update for a workstation. Some software updates require a system restart to complete the installation process.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Title<String[]>

Specifies an array of criteria, as strings, for software updates. The rule adds software updates that have titles that meet specified criteria to the software update group.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UpdateClassification<String[]>

Specifies an array of criteria, as strings, for software updates. The rule adds software updates that have update classifications that meet specified criteria to the software update group.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UpdateDescription<String[]>

Specifies an array of criteria, as strings, for software updates. The rule adds software updates that have update descriptions that meet specified criteria to the software update group.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UseBranchCache<Boolean>

Indicates whether to use a branch cache for this update deployment. If you specify a value of \$True, clients share content on the same subnet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UserNotification<UserNotificationType>

Specifies the type of user notification. Valid values are:

- DisplayAll. Display in Software Center and show all notifications.
- DisplaySoftwareCenterOnly. Display in Software Center, and only show notifications of computer restarts.
- HideAll. Hide in Software Center and all notifications.

The acceptable values for this parameter are:

DisplayAll	
DisplaySoftwareCenterOnly	
HideAll	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UseUtc<Boolean>

Indicates whether to use Coordinated Universal Time (UTC), also known as Greenwich Mean Time. If this value is \$True, Configuration Manager uses UTC for this deployment. If this value is \$False, Configuration Manager uses local time.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-VerboseLevel<VerboseLevelType>

Specifies the level of detail you want clients to report for deployments that this rule creates. Valid values are:

- AllMessages
- OnlyErrorMessage
- OnlySuccessAndErrorMessage

The acceptable values for this parameter are:

AllMessages	
OnlyErrorMessage	
OnlySuccessAndErrorMessage	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-WriteFilterHandling<Boolean>

Indicates whether to enable write filters for embedded devices. For a value of \$True, the device commits changes during a maintenance window. This action requires a restart. For a value of \$False, the device saves changes in an overlay and commits them later.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Create an automatic deployment rule

This command creates a deployment rule named DeploymentRule07 for the collection named Desktops and the deployment package named Updates123. The rule deploys updates that have an article ID that contains 117.

```
PS C:\> New-CMSoftwareUpdateAutoDeploymentRule -CollectionName "Desktops" -
DeploymentPackageName "Updates123" -Name "DeploymentRule07" -ArticleId "117"
```

Example 2: Create an automatic deployment rule that uses a schedule

This example creates an automatic deployment rule that uses a defined schedule. Deployment takes place according to the schedule.

The first command creates a schedule for the Wednesday day of the week, and stores the schedule object in the \$Schedule variable. For more information, see help for the **New-CMSchedule** cmdlet.

The second command creates an automatic deployment rule for updates that uses the schedule object stored in the \$Schedule variable. This command specifies values for a large number of parameters.

```
PS C:\> $Schedule = New-CMSchedule -DayOfWeek Wednesday
PS C:\> New-CMSoftwareUpdateAutoDeploymentRule -CollectionName "Laptops" -
```

```
DeploymentPackageName "Updates235" -Name "DeploymentRule22" -
AddToExistingSoftwareUpdateGroup $False -AlertTime 4 -AlertTimeUnit Weeks -AllowRestart
$True -AllowSoftwareInstallationOutsideMaintenanceWindow $True -AllowUseMeteredNetwork $True
-ArticleId "test" -AvailableImmediately $False -AvailableTime 5 -AvailableTimeUnit Months -
CustomSeverity Critical -DateReleasedOrRevised Last1day -DeadlineImmediately $False -
DeadlineTime $True -DeadlineTimeUnit Hours -DeployWithoutLicense $True -Description
"Standard updates for our laptop systems." -DisableOperationManager $True -
DownloadFromInternet $False -DownloadFromMicrosoftUpdate $True -EnabledAfterCreate $False -
GenerateOperationManagerAlert $True -GenerateSuccessAlert $True -Language "Catalan" -
LanguageSelection "English" -Location "\\k\as_015_Client_Dev_1" -MicrosoftAsVendor $True -
NoInstallOnRemote $False -NoInstallOnUnprotected $True -RunType RunTheRuleOnSchedule -
Schedule $Schedule -SendWakeUpPacket $True -SuccessPercent 99 -Superseded $True -
SuppressRestartServer $True -SuppressRestartWorkstation $True -UpdateClassification
"Critical Updates" -UseBranchCache $False -UserNotification DisplayAll -UseUtc $True -
VerboseLevel AllMessages -WriteFilterHandling $True
```

Related topics

[Disable-CMSoftwareUpdateAutoDeploymentRule](#)

[Enable-CMSoftwareUpdateAutoDeploymentRule](#)

[Get-CMSoftwareUpdateAutoDeploymentRule](#)

[Invoke-CMSoftwareUpdateAutoDeploymentRule](#)

[Remove-CMSoftwareUpdateAutoDeploymentRule](#)

[Set-CMSoftwareUpdateAutoDeploymentRule](#)

[New-CMSchedule](#)

New-CMSoftwareUpdateGroup

New-CMSoftwareUpdateGroup

Creates a software update group.

Syntax

Parameter Set: New

```
New-CMSoftwareUpdateGroup -Name <String> -UpdateId <Int32[]> [-Description <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **New-CMSoftwareUpdateGroup** cmdlet creates a software update group for Microsoft System Center 2012 Configuration Manager. A software update group is a collection of one or more software updates. You can add software updates to a software update group and then deploy the group to clients. After you deploy a software update group, you can add new software updates to the group and System Center 2012 Configuration Manager automatically deploys them.

Parameters

-Description<String>

Specifies a description of a software update group.

Aliases	LocalizedDescription
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies a name of a software update group.

Aliases	LocalizedDisplayName
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UpdateId<Int32[]>

Specifies an array of IDs of software updates.

Aliases	Updates
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Create a software update group

This command creates a software update group named ClientUpdateGroup01 that includes the software update that has the update ID 100027.

```
PS C:\> New-CMSoftwareUpdateGroup -Name "ClientUpdateGroup01" -UpdateID 100027 -Description "Client software update group 01 for Accounts Payable"
```

Related topics

[Get-CMSoftwareUpdateGroup](#)

[Remove-CMSoftwareUpdateGroup](#)

[Set-CMSoftwareUpdateGroup](#)

New-CMSqlServerSetting

New-CMSqlServerSetting

Creates a SQL Server settings object in Configuration Manager.

Syntax

Parameter Set: NewSqlServerSettingsbyCopy

```
New-CMSqlServerSetting -CopySqlServerExpressOnSecondarySite [-SqlServerServiceBrokerPort <Int32> ] [-SqlServerServicePort <Int32> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: NewSqlServerSettingsbyExisting

```
New-CMSqlServerSetting -SiteDatabaseName <String> -UseExistingSqlServerInstance [- InstanceName <String> ] [-SqlServerServiceBrokerPort <Int32> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **New-CMSqlServerSetting** cmdlet creates a Microsoft SQL Server settings object in Microsoft System Center 2012 Configuration Manager. The object specifies settings for the name of the site database and the port number for the SQL Server service and SQL Server Service Broker.

Parameters

-CopySqlServerExpressOnSecondarySite

Indicates that Microsoft SQL Server Express is copied to a secondary site.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InstanceName<String>

Specifies the name of an instance of SQL Server.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteDatabaseName<String>

Specifies a name of the Configuration Manager site database.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SqlServerServiceBrokerPort<Int32>

Specifies a port number for the SQL Server Service Broker port.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SqlServerServicePort<Int32>

Specifies a port number for the SQL Server service port.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UseExistingSqlServerInstance

Indicates that you use the existing instance of SQL Server.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Create a SQL Server settings object

This command creates a SQL Server settings object that specifies that System Center 2012 Configuration Manager copies Microsoft SQL Server Express to a secondary site. The command specifies that System Center 2012 Configuration Manager use port 4037 for the SQL Server Service Broker.

```
PS C:\> New-CMSqlServerSetting -CopySqlServerExpressOnSecondarySite -  
SqlServerServiceBrokerPort 4037
```

New-CMStatusFilterRule

New-CMStatusFilterRule

Creates a rule in Configuration Manager.

Syntax

Parameter Set: Default

```
New-CMStatusFilterRule -Name <String> -SiteCode <String> [-  
AllowUserDeleteMessagesAfterThresholdDays <Int32> ] [-ComponentName <String> ] [-  
ForwardToStatusSummarizer <Boolean> ] [-MessageId <String> ] [-MessageType {Audit | Detail |  
Milestone | None} ] [-ProcessLowerPriorityRule <Boolean> ] [-ProgramPath <String> ] [-  
PropertyId <String> ] [-PropertyValue <String> ] [-ReplicateToParentSite <Boolean> ] [-  
ReplicationPriority {High | Low | Medium} ] [-ReportToEventLog <Boolean> ] [-RunProgram  
<Boolean> ] [-SeverityType {Error | Informational | None | Warning} ] [-SiteSystemServerName  
<String> ] [-Source <String> ] [-StatusFilterRuleSiteCode <String> ] [-WriteToDatabase  
<Boolean> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **New-CMStatusFilterRule** cmdlet creates a rule that triggers one or more actions that alerts an administrator to a specific message in Microsoft System Center 2012 Configuration Manager.

Parameters

-AllowUserDeleteMessagesAfterThresholdDays<Int32>

Specifies how long, in days, to retain messages. Set the *WriteToDatabase* parameter to enable this setting.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ComponentName<String>

Specifies the Configuration Manager component that corresponds to the status messages.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ForwardToStatusSummarizer<Boolean>

Indicates whether to forward to the status summarizer.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-MessageId<String>

Specifies a message ID in Configuration Manager.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-MessageType<MessageType>

Specifies a status message type in Configuration Manager. Valid values are: Audit, Detail, Milestone, and None.

The acceptable values for this parameter are:

Audit	
Detail	
Milestone	
None	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies a name for the status filter rule.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ProcessLowerPriorityRule<Boolean>

Indicates whether to process a lower priority rule, which prevents further rule processing.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ProgramPath<String>

Specifies a path to a program that runs when a status message matches the status filter rule.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PropertyId<String>

Specifies a property ID in Configuration Manager.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PropertyValue<String>

Specifies a value for the corresponding *PropertyId* parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ReplicateToParentSite<Boolean>

Indicates whether to pass a message to the parent site.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ReplicationPriority<ReplicationPriority>

Specifies a replication priority for sending status messages to the parent site. Valid values are: High, Low, and Medium.

The acceptable values for this parameter are:

High	
Low	
Medium	

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ReportToEventLog<Boolean>

Indicates whether to report an event in the Windows event log.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RunProgram<Boolean>

Indicates whether to run a program when a status message matches a filter rule.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SeverityType<SeverityType>

Specifies the severity of a status message. Valid values are: Error, Informational, None, and Warning.

The acceptable values for this parameter are:

Error	
Informational	
None	

Warning	
---------	--

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteCode<String>

Specifies a System Center 2012 Configuration Manager site code that defines the status rule.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteSystemServerName<String>

Specifies a name of the site system server.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Source<String>

Specifies the status message source to match.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-StatusFilterRuleSiteCode<String>

Specifies a site code for the status filter rule.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-WriteToDatabase<Boolean>

Indicates whether to write a message to the database. Must be set to enable the *AllowUserDeleteMessagesAfterThresholdDays* parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Create a new status filter rule

This command creates a new status filter rule.

```
PS C:\> New-CMStatusFilterRule -SiteCode "Contoso2" -Name "11211" -TargetSiteCode "Contoso2"
-Source "SMS Provider" -System "Contoso-test" -Component "1111" -MessageType Milestone -
SeverityType Informational -MessageId "1234" -PropertyName "403" -PropertyValue "123" -
WriteToDatabase $True -AllowDeleteMessagesTimeDays 20 -ReportToEventLog $False -
ReplicateToParentSite $False -RunProgram $False -ForwardToStatusSummarizers $True -
ProcessLowerPriorityRules $True
```

Related topics

[Disable-CMStatusFilterRule](#)

[Enable-CMStatusFilterRule](#)

[Get-CMStatusFilterRule](#)

[Remove-CMStatusFilterRule](#)

[Set-CMStatusFilterRule](#)

New-CMStatusMessageQuery

New-CMStatusMessageQuery

Creates a status message query.

Syntax

Parameter Set: NewStatusMessageQuery

```
New-CMStatusMessageQuery -Name <String> [-Comments <String> ] [-Expression <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **New-CMStatusMessageQuery** cmdlet creates a status message query in Microsoft System Center 2012 Configuration Manager. Status message queries in System Center 2012 Configuration Manager return status messages from the site database. All major System Center 2012 Configuration Manager components generate status messages.

Parameters

-Comments<String>

Specifies a description for the status message query.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Expression<String>

Specifies WMI Query Language (WQL) text for the query.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies a name for the status message query.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Create a status message query

This command creates a status message query named Client Component Configuration Changes and Fatal Errors. The *Expression* parameter specifies the WMI Query Language (WQL) text for the query.

```
PS C:\> New-CMStatusMessageQuery -Name "Client Component Configuration Changes and Fatal Errors" -Expression "select stat.*, ins.*, att1.*, stat.Time from SMS_StatusMessage as stat left join SMS_StatMsgInsStrings as ins on stat.RecordID = ins.RecordID left join SMS_StatMsgAttributes as att1 on stat.RecordID = att1.RecordID where stat.ModuleName = 'SMS Client' and stat.MessageID = 669 and stat.SiteCode = ##PRM:SMS_StatusMessage.SiteCode## and stat.Time >= ##PRM:SMS_StatusMessage.Time## order by stat.Time desc"
```

Related topics

[Get-CMStatusMessageQuery](#)

[Set-CMStatusMessageQuery](#)

[Remove-CMStatusMessageQuery](#)

New-CMStoragefolder

New-CMStoragefolder

Creates a new storage folder in Configuration Manager.

Syntax

Parameter Set: NewStorageFolder

```
New-CMStoragefolder -StorageFolderName <String> [-MaximumClientNumber <Int32> ] [-MinimumFreeSpace <Int32> ] [-SpaceUnit {Gigabyte | Megabyte | Percent} ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **New-CMStoragefolder** cmdlet creates a new storage folder to store user migration data in Microsoft System Center 2012 Configuration Manager.

A storage folder identifies a location on a state migration point site system to store user migration data. Use this cmdlet in conjunction with the **Add-CMStateMigrationPoint** cmdlet to create a new state migration point with storage folders.

Parameters

-MaximumClientNumber<Int32>

Specifies the maximum number of clients that the storage folder can hold. The storage folder contains user state migration data in Configuration Manager. Valid values are: numbers between 1 and 99999.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-MinimumFreeSpace<Int32>

Specifies the minimum amount of free space for storage of user state migration data. Valid values are: numbers between 1 - 99999 when specifying a byte value, or numbers between 1 - 100 when specifying a percentage.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SpaceUnit<MinSpaceType>

Specifies the storage units for the *MinimumFreeSpace* parameter.

The acceptable values for this parameter are:

Gigabyte	
Megabyte	
Percent	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-StorageFolderName<String>

Specifies a local path for the storage folder. The associated state migration point site system role in Configuration Manager uses this path.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Create a new storage folder

This command creates a new storage folder for migration data by using the maximum number of clients, minimum free space, and storage folder path parameters.

```
PS C:\> New-CMStoragefolder -MaximumClientNumber 80 -MinimumFreeSpace 10 -SpaceUnit Megabyte  
-StorageFolderName "D:\Contoso-Mobile-Users"
```

New-CMTaskSequence

New-CMTaskSequence

Creates a task sequence.

Syntax

Parameter Set: NewBuildOSImage

```
New-CMTaskSequence -BootImagePackageId <String> -BuildOperatingSystemImageOption -JoinDomain {DomainType | WorkgroupType} -OperatingSystemFileAccount <String> -OperatingSystemFilePath <String> -OperatingSystemImageIndex <UInt32> -OperatingSystemImagePackageId <String> -TaskSequenceName <String> [-ApplicationName <String[]> ] [-ApplyAllImages <Boolean> ] [-ClientPackagePackageId <String> ] [-CreatedBy <String> ] [-DomainAccount <String> ] [-DomainName <String> ] [-DomainOrganizationUnit <String> ] [-DomainPassword <SecureString> ] [-GeneratePassword <Boolean> ] [-IgnoreInvalidApplication <Boolean> ] [-ImageDescription <String> ] [-ImageVersion <String> ] [-InstallationLicensingMode {NonSpecify | PerSeat | PerServer} ] [-InstallationProductkey <String> ] [-InstallationProperty <String> ] [-LocalAdminPassword <SecureString> ] [-MaximumServerConnection <Int32> ] [-OperatingSystemFileAccountPassword <SecureString> ] [-SoftwareUpdateStyle {All | Mandatory | NoInstall} ] [-SystemPreparationPackageId <String> ] [-TaskSequenceDescription <String> ] [-WorkgroupName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: NewCustom

```
New-CMTaskSequence -CustomOption -TaskSequenceName <String> [-BootImagePackageId <String> ] [-TaskSequenceDescription <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: NewInstallOSImage

```
New-CMTaskSequence -BootImagePackageId <String> -InstallOperatingSystemImageOption -JoinDomain {DomainType | WorkgroupType} -OperatingSystemImageIndex <UInt32> -OperatingSystemImagePackageId <String> -TaskSequenceName <String> [-ApplicationName <String[]> ] [-ApplyAllImages <Boolean> ] [-CaptureLocallyUsingLinks <Boolean> ] [-CaptureNetworkSetting <Boolean> ] [-CaptureUserSetting <Boolean> ] [-CaptureWindowsSetting <Boolean> ] [-ClientPackagePackageId <String> ] [-ConfigureBitLocker <Boolean> ] [-DomainAccount <String> ] [-DomainName <String> ] [-DomainOrganizationUnit <String> ] [-DomainPassword <SecureString> ] [-GeneratePassword <Boolean> ] [-IgnoreInvalidApplication <Boolean> ] [-InstallationLicensingMode {NonSpecify | PerSeat | PerServer} ] [-InstallationProductkey <String> ] [-InstallationProperty <String> ] [-LocalAdminPassword <SecureString> ] [-PartitionAndFormatTarget <Boolean> ] [-SaveLocally <Boolean> ] [-SoftwareUpdateStyle {All | Mandatory | NoInstall} ] [-TaskSequenceDescription <String> ] [-UserStateMigrationToolPackageId <String> ] [-WorkgroupName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: NewInstallOSImageVhd

```
New-CMTaskSequence -BootImagePackageId <String> -InstallOperatingSystemImageVhdOption -JoinDomain {DomainType | WorkgroupType} -OperatingSystemImageIndex <UInt32> -OperatingSystemImagePackageId <String> -TaskSequenceName <String> [-ApplicationName <String[]> ] [-ApplyAllImages <Boolean> ] [-ClientPackagePackageId <String> ] [-
```

```
ConfigureBitLocker <Boolean> ] [-DomainAccount <String> ] [-DomainName <String> ] [-DomainOrganizationUnit <String> ] [-DomainPassword <SecureString> ] [-GeneratePassword <Boolean> ] [-IgnoreInvalidApplication <Boolean> ] [-InstallationLicensingMode {NonSpecify | PerSeat | PerServer} ] [-InstallationProductkey <String> ] [-InstallationProperty <String> ] [-LocalAdminPassword <SecureString> ] [-PartitionAndFormatTarget <Boolean> ] [-TaskSequenceDescription <String> ] [-WorkgroupName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **New-CMTaskSequence** cmdlet creates a Microsoft System Center 2012 Configuration Manager task sequence. A task sequence performs multiple steps or tasks on a System Center 2012 Configuration Manager client computer without user intervention.

Parameters

-ApplicationName<String[]>

Specifies an array of names of applications included in the task sequence.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ApplyAllImages<Boolean>

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-BootImagePackageId<String>

Specifies the ID of the boot image package associated with the task sequence.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-BuildOperatingSystemImageOption

Indicates that the task sequence builds and captures a reference operating system image from a set of operating system installation files.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CaptureLocallyUsingLinks<Boolean>

Indicates whether Configuration Manager stores the captured data locally on the destination computer. The links that Configuration Manager uses to store the user state locally are referred to as hard-links. Hard-links is a User State Migration Tool (USMT) 4.0 feature that scans the computer for user files and settings and then creates a directory of hard-links to those files. The hard-links are then used to restore the user data after the new operating system is deployed.

Aliases	none
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CaptureNetworkSetting<Boolean>

Indicates whether the task sequence captures network settings from the computer that runs the task sequence.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CaptureUserSetting<Boolean>

Indicates whether the task sequence captures the user state. If you specify this parameter, specify the *UserStateMigrationToolPackageId* parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CaptureWindowsSetting<Boolean>

Indicates whether the task sequence captures Windows settings from the computer that runs the task sequence. You can capture the computer name, registered user and organization name, and the time zone settings.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ClientPackagePackageId<String>

Specifies the ID of the Configuration Manager client package to install on the destination computer.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ConfigureBitLocker<Boolean>

Indicates whether the task sequence enables BitLocker encryption on the hard drive.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CreatedBy<String>

Specifies the name of the user that created the operating system image that the task sequence captures.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CustomOption

Indicates that the cmdlet creates a custom task sequence.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DomainAccount<String>

Specifies an account, in the format *Domain\User*, that has permissions necessary to join the computer to the domain.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DomainName<String>

Specifies a domain name. Include this parameter to have the target computer join the specified domain.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DomainOrganizationUnit<String>

Specifies the Lightweight Directory Access Protocol (LDAP) path for an organizational unit (OU) for the computer to join. Use the following format: LDAP//OU=*computers*, DC=*Contoso.com*, C=*com*. Specify an OU in the domain that you specified in the *DomainName* parameter.

If the computer is already a member of some other OU, Active Directory Domain Services (AD DS) does not allow you to change the OU and this setting is ignored.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DomainPassword<SecureString>

Specifies the password, as a secure string, for the user account that you specified for the *DomainAccount* parameter.

Aliases	none
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-GeneratePassword<Boolean>

Indicates whether Configuration Manager randomly generates a password for the local administrator account in the new operating system.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-IgnoreInvalidApplication<Boolean>

Indicates whether the task sequence step continues if an individual application installation fails.

If you specify this parameter, the task sequence continues regardless of any installation errors. If you do not specify this parameter, the task sequence step will end immediately when an installation fails.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ImageDescription<String>

Specifies a description of the operating system image that the task sequence captures.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ImageVersion<String>

Specifies the version of the operating system that the task sequence captures.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InstallationLicensingMode<ServerLicensingMode>

Specifies the Windows Server license mode that the task sequence uses. Valid values are:

- NonSpecify
- PerSeat
- PerServer

The acceptable values for this parameter are:

NonSpecify	
PerSeat	
PerServer	

Aliases	none
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InstallationProductkey<String>

Specifies the Windows product key.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InstallationProperty<String>

Specifies Configuration Manager client installation properties.

Site assignment and the default configuration are automatically specified by the task sequence action. You can use this field to specify any additional installation properties to use when you install the client. To enter multiple installation properties, separate them with a space. If a property contains spaces, surround it by quotation marks (").

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InstallOperatingSystemImageOption

Indicates that the task sequence installs an existing image option.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InstallOperatingSystemImageVhdOption

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-JoinDomain<JoinType>

Specifies the destination computer to add to a workgroup or domain. Valid values are:

- DomainType
- WorkgroupType

The acceptable values for this parameter are:

DomainType	
WorkgroupType	

Aliases	none
Required?	true
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-LocalAdminPassword<SecureString>

Specifies the local administrator password, as a secure string, for the destination computer.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-MaximumServerConnection<Int32>

Specifies the maximum number of server connections. Specify this parameter if you specify PerServer for the *InstallationLicensingMode* parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OperatingSystemFileAccount<String>

Specifies the Windows account that has permissions to the network share that you specify in the *OperatingSystemFilePath* parameter.

Aliases	none
Required?	true

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OperatingSystemFileAccountPassword<SecureString>

Specifies the password, as a secure string, for the account that you specify in the *OperatingSystemFileAccount* parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OperatingSystemFilePath<String>

Specifies the file system pathname to the location that Configuration Manager uses when it stores the captured operating system image.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OperatingSystemImageIndex<UInt32>

Specifies the index of the operating system image to install. Specify this parameter if the operating system image package has multiple images.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OperatingSystemImagePackageId<String>

Specifies the ID of the package that contains the operating system image to install.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PartitionAndFormatTarget<Boolean>

Indicates whether the task sequence partitions and formats the destination computer before the operating system is installed.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SaveLocally<Boolean>

Indicates whether Configuration Manager clients download the full image to a local cache before the task sequence runs. This parameter has been deprecated.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SoftwareUpdateStyle<SoftwareUpdateStyleType>

Specifies whether the task sequence installs all updates or only mandatory updates for the destination computers that receive the task sequence. Valid values are:

- All
- Mandatory
- NoInstall

The acceptable values for this parameter are:

All	
Mandatory	
NoInstall	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SystemPreparationPackageId<String>

Specifies the ID of the Configuration Manager package that contains the appropriate version of Sysprep to use to capture the reference computer settings.

If the operating system version that you are running is Windows Vista or later, Sysprep is automatically installed on the computer and you do not have to specify a package. If the operating system version

that you are running is Windows XP SP3 or Windows Server 2003 SP2, you must specify a package that contains the version of Sysprep and its support files that is appropriate for that operating system version. This package does not require a program. Configuration Manager uses the Sysprep files contained in the package.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-TaskSequenceDescription<String>

Specifies a description for the task sequence.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-TaskSequenceName<String>

Specifies a name for the task sequence.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UserStateMigrationToolPackageId<String>

Specifies the ID of the USMT package.

To store the user state data locally or on a state migration point, you must create a package that contains the USMT source files that you want to use.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-WorkgroupName<String>

Specifies the name of a workgroup. Specify this parameter if you specified *WorkgroupType* for the *JoinDomain* parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Create a task sequence that installs the operating system from a boot image

This command creates a task sequence named TaskSeq01 that uses the boot image package that has the ID CM200002.

The command specifies the *CustomOption* parameter to indicate a custom task sequence.

```
PS C:\> New-CMTaskSequence -CustomOption -BootImagePackageID "CM200002" -TaskSequenceName "TaskSeq01" -TaskSequenceDescription "Custom task sequence 22"
```

Example 2: Create a task sequence that builds the operating system image

This command creates a task sequence named BuildOStask01. The task sequence uses the boot image package that has the ID CM200002 to build an operating system image from a set of operating system installation files in the operating system image package that has the ID CM20000D.

```
PS C:\> New-CMTaskSequence -BuildOperatingSystemImageOption -BootImagePackageId "CM200002" -TaskSequenceName "BuildOStask01" -OperatingSystemImagePackageId "CM20000D" -OperatingSystemImageIndex "1" -JoinDomain WorkgroupType -WorkgroupName "Midwest" -OperatingSystemFilePath "\\btc-dist-08\Public\CM\AdminTeam\OpSys01\nodesc_boot.wim" -OperatingSystemFileAccount "Contoso\Primeos" -OperatingSystemFileAccountPassword "Basket11"
```

Example 3: Create a task sequence that installs the operating system image

This command creates a task sequence that uses the boot image package that has the ID CM200002 to install an operating system image from the operating system image package that has the ID CM20000D.

```
PS C:\> New-CMTaskSequence -InstallOperatingSystemImageOption -BootImagePackageId "CM200002" -TaskSequenceName "InstallOStask01" -OperatingSystemImagePackageId "CM20000D" -OperatingSystemImageIndex "1" -JoinDomain WorkgroupType -WorkgroupName "Midwest" -UserStateMigrationToolPackageID "CM200001" -InstallationProductkey "TWB3K-HKMBK-HXRBD-TQGK9-VJRMG"
```

Related topics

[Enable-CMTaskSequence](#)

[Get-CMTaskSequence](#)

[Import-CMTaskSequence](#)

[Set-CMTaskSequence](#)

[Disable-CMTaskSequence](#)

[Remove-CMTaskSequence](#)

[New-CMTaskSequenceMedia](#)

New-CMTaskSequenceMedia

New-CMTaskSequenceMedia

Creates task sequence media in System Center 2012 Configuration Manager.

Syntax

Parameter Set: NewBootableMedia

```
New-CMTaskSequenceMedia -BootableMediaOption -BootImageId <String> -
DistributionPointServerName <String> -ManagementPointServerName <String[]> -MediaInputType
{CDDVD | USB} -MediaMode {Dynamic | SiteBased} -ProtectPassword <Boolean> [-
AllowUnattendedDeployment <Boolean> ] [-CommandDistributionPointServerName <String> ] [-
CommandIncludeFile <Boolean> ] [-Commandline <String> ] [-CommandPackageName <String> ] [-
CreateMediaSelfCertificate <Boolean> ] [-DriveName <String> ] [-EnablePrestartCommand
<Boolean> ] [-EnableUnknownSupport <Boolean> ] [-ExpirationDate <DateTime> ] [-
ImportCertificatePassword <SecureString> ] [-ImportCertificatePath <String> ] [-MediaPath
<String> ] [-MediaSize {None | Size4GB | Size650MB | Size8GB | SizeUnlimited} ] [-Password
<SecureString> ] [-StartDate <DateTime> ] [-UserDeviceAffinity {AdministratorApproval |
AutoApproval | DoNotAllow} ] [-Variable <Hashtable> ] [-Confirm] [-WhatIf] [
<CommonParameters>]
```

Parameter Set: NewCaptureMedia

```
New-CMTaskSequenceMedia -BootImageId <String> -CaptureMediaOption -
DistributionPointServerName <String> -MediaInputType {CDDVD | USB} [-DriveName <String> ] [-
MediaPath <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: NewPrestagedMedia

```
New-CMTaskSequenceMedia -BootImageId <String> -DistributionPointServerName <String> -
ManagementPointServerName <String[]> -MediaMode {Dynamic | SiteBased} -
OperatingSystemImageDistributionPointServerName <String> -PrestagedMediaOption -
ProtectPassword <Boolean> -TaskSequenceDistributionPointServerName <String[]> -
TaskSequenceId <String> [-AllowUnattendedDeployment <Boolean> ] [-ApplicationName <String[]>
] [-Comment <String> ] [-CreatedBy <String> ] [-MediaPath <String> ] [-
OperatingSystemImageName <String> ] [-OperatingSystemImagePackageId <String> ] [-
PackageDriverName <String[]> ] [-PackageName <String[]> ] [-Variable <Hashtable> ] [-Version
<String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: NewStandAloneMedia

```
New-CMTaskSequenceMedia -MediaInputType {CDDVD | USB} -ProtectPassword <Boolean> -
StandAloneMediaOption -TaskSequenceDistributionPointServerName <String[]> -TaskSequenceId
<String> [-AllowUnattendedDeployment <Boolean> ] [-CommandDistributionPointServerName
<String> ] [-CommandIncludeFile <Boolean> ] [-Commandline <String> ] [-CommandPackageName
<String> ] [-DriveName <String> ] [-EnablePrestartCommand <Boolean> ] [-
IncludeApplicationDependencies <Boolean> ] [-MediaPath <String> ] [-MediaSize {None |
Size4GB | Size650MB | Size8GB | SizeUnlimited} ] [-Password <SecureString> ] [-Variable
<Hashtable> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **New-CMTaskSequenceMedia** cmdlet creates task sequence media in Microsoft System Center 2012 Configuration Manager.

Parameters

-AllowUnattendedDeployment<Boolean>

Indicates whether you allow unattended operating system deployment, which does not prompt for network configuration or optional task sequences.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ApplicationName<String[]>

Specifies an array of names of applications included in the task sequence.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-BootableMediaOption

Indicates that bootable media is an option. Bootable media lets you deploy an operating system when the destination computer starts.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-BootImageId<String>

Specifies the ID of the boot image package associated with the task sequence media.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CaptureMediaOption

Indicates that the task sequences are written to capture media. Capture media contains an operating system image from a reference computer, including a boot image that starts a reference computer, and the task sequence that captures the operating system.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CommandDistributionPointServerName<String>

Specifies a name for a distribution point server from which the cmdlet acquires the package. The *CommandPackageName* parameter specifies the package name.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CommandIncludeFile<Boolean>

Indicates whether to include a file.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Commandline<String>

Specifies a command line for a task sequence.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-CommandPackageName<String>

Specifies a package name for the command specified by the *CommandLine* parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Comment<String>

Specifies a comment for a prestaged media file.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CreatedBy<String>

Specifies the name of an individual or organization responsible for the creation of the prestaged media.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-CreateMediaSelfCertificate<Boolean>

Indicates whether the media includes a self-signed certificate. Use this parameter only in mixed-mode environments.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DistributionPointServerName<String>

Specifies a name for a distribution point server.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DriveName<String>

Specifies a drive name.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnablePrestartCommand<Boolean>

Indicates whether to enable a prestart command. A prestart command is a script or executable that runs before the task sequence.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnableUnknownSupport<Boolean>

Indicates whether to provision unknown systems for operating system deployment.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ExpirationDate<DateTime>

Specifies an expiration date, in D.HH:MM:SS format, for bootable media.

Aliases	none
Required?	false
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ImportCertificatePassword<SecureString>

Specifies a password for an import certificate, as a secure string. An import certificate is a PKI-issued certificate added to the boot media for client authentication and communication with a System Center 2012 Configuration Manager site.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ImportCertificatePath<String>

Specifies a path for an import certificate to add to the boot media.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-IncludeApplicationDependencies<Boolean>

Aliases	none
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ManagementPointServerName<String[]>

Specifies an array of names for management point servers.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-MediaInputType<MediaInputType>

Specifies a media input type. Valid values are:

-- CDDVD

-- USB

The acceptable values for this parameter are:

CDDVD	
USB	

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-MediaMode<MediaMode>

Specifies a media mode. Valid values are:

- Dynamic
- SiteBased

The acceptable values for this parameter are:

Dynamic	
SiteBased	

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-MediaPath<String>

Specifies a path to the media.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-MediaSize<MediaSize>

Specifies the size of the media. Valid values are:

- None

- Size4GB
- Size650MB
- Size8GB
- SizeUnlimited

The acceptable values for this parameter are:

None	
Size4GB	
Size650MB	
Size8GB	
SizeUnlimited	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OperatingSystemImageDistributionPointServerName<String>

Specifies the name of a distribution point server for an operating system image.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OperatingSystemImageName<String>

Specifies the name of an operating system image.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OperatingSystemImagePackageId<String>

Specifies the identifier of an operating system image package.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PackageDriverName<String[]>

Specifies an array of package driver names.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PackageName<String[]>

Specifies an array of package names.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Password<SecureString>

Specifies a password, as a secure string.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PrestagedMediaOption

Indicates that the bootable media image and operating system is written to a hard disk. Prestaged media contains a boot image and the operating system to apply to a destination computer.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ProtectPassword<Boolean>

Indicates whether to protect the media with a password.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-StandAloneMediaOption

Indicates that the task sequences are written to standalone media, such as DVD or CD.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-StartDate<DateTime>

Specifies a start date and time, in D.HH:MM:SS format.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-TaskSequenceDistributionPointServerName<String[]>

Specifies an array of available distribution point servers for a task sequence.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-TaskSequenceId<String>

Specifies an ID for a task sequence.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UserDeviceAffinity<UserDeviceAffinityType>

Specifies user device affinity. User device affinity associates users with a destination computer. Valid values are:

- AdministratorApproval
- AutoApproval
- DoNotAllow

The acceptable values for this parameter are:

AdministratorApproval	
AutoApproval	
DoNotAllow	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Variable<Hashtable>

Specifies a task sequence variable. The task sequence variable consists of a name and a value.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Version<String>

Specifies the version information for the media.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Create task sequence media with the captured media option

The command uses the **New-CMTaskSequenceMedia** cmdlet to create task sequence media by using the *CaptureMediaOption* parameter. The command also specifies a value for the *MediaPath* parameter, and a value for the *MediaInputType* parameter.

```
PS C:\> New-CMTaskSequenceMedia -CaptureMediaOption -MediaPath
"\\Contoso320\Users\Administrator.Contoso319DOM\Desktop\DD\1.iso" -MediaInputType CDDVD -
BootImageName "Boot" -DistributionPointServerName "Contoso320.Contoso319DOM.NET"
```

Example 2: Create task sequence media with the standalone media option

The first command creates a mapping and stores the result in the \$Group variable.

The second command uses the **New-CMTaskSequenceMedia** cmdlet to create task sequence media by using the *StandAloneMediaOption* parameter.

```
PS C:\> $Group = @{"6"="8"};PS C:\> New-CMTaskSequenceMedia -StandAloneMediaOption -Variable $Group -MediaInputType CDDVD -MediaPath "\\Contoso320\Users\Administrator.Contoso319DOM\Desktop\DD\111 - Copy.iso" -ProtectPassword 0 -TaskSequenceId "CCC0000B" -TaskSequenceDistributionPointServerName "\\Contoso320.Contoso319DOM.NET"
```

Example 3: Create task sequence media with the bootable media option

The command uses the **New-CMTaskSequenceMedia** cmdlet to create task sequence media by using the *BootableMediaOption* parameter.

```
PS C:\> New-CMTaskSequenceMedia -BootableMediaOption -MediaInputType CDDVD -MediaPath "\\Contoso320\Users\Administrator.Contoso319DOM\Desktop\DD\111 - Copy (6).iso" -MediaMode Dynamic -ProtectPassword 0 -BootImageName "boot" -DistributionPointServerName "Contoso320.Contoso319DOM.NET" -ManagementnPointNetworkOperatingSystemPath "Contoso320.Contoso319DOM.NET"
```

Example 4: Create task sequence media with the prestaged media option

The command uses the **New-CMTaskSequenceMedia** cmdlet to create task sequence media by using the *PrestagedMediaOption* parameter.

```
PS C:\> New-CMTaskSequenceMedia -PrestagedMediaOption -MediaMode Dynamic -MediaPath "\\Contoso320\Users\Administrator.Contoso319DOM\Desktop\DD\2.wim" -ProtectPassword 0 -TaskSequenceId "CCC0000B" -BootImageName "boot" -DistributionPointServerName "Contoso320.Contoso319DOM.NET" -ManagementnPointNetworkOperatingSystemPath "Contoso320.Contoso319DOM.NET" -OperatingSystemImageDistributionPointServerName "Contoso320.Contoso319DOM.NET" -TaskSequenceDistributionPointServerName "\\Contoso320.Contoso319DOM.NET"
```

Related topics

[Get-CMTaskSequence](#)

New- CMTrustedRootCertificateProfileConfigurationItem

New-CMTrustedRootCertificateProfileConfigurationItem

Creates a root certificate profile.

Syntax

Parameter Set: Default

```
New-CMTrustedRootCertificateProfileConfigurationItem -DesiredConfigurationDigestPath  
<String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **New-CMTrustedRootCertificateProfileConfigurationItem** cmdlet creates a root certificate profile. Client computers use root certificate profiles to chain their certificates back to a corporate public key infrastructure (PKI) certification authority.

Parameters

-DesiredConfigurationDigestPath<String>

Specifies a path to the configuration data stored as a digest.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

[Copy-CMTrustedRootCertificateProfileConfigurationItem](#)

[Get-CMTrustedRootCertificateProfileConfigurationItem](#)

[Remove-CMTrustedRootCertificateProfileConfigurationItem](#)

[Set-CMTrustedRootCertificateProfileConfigurationItem](#)

New-CMUserCollection

New-CMUserCollection

Creates a collection for users and adds the collection to the Configuration Manager hierarchy.

Syntax

Parameter Set: NewByLimitName

```
New-CMUserCollection -LimitingCollectionName <String> -Name <String> [-Comment <String> ] [-RefreshSchedule <IResultObject> ] [-RefreshType {Both | ConstantUpdate | Manual | Periodic} ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: NewByLimitId

```
New-CMUserCollection -LimitingCollectionId <String> -Name <String> [-Comment <String> ] [-RefreshSchedule <IResultObject> ] [-RefreshType {Both | ConstantUpdate | Manual | Periodic} ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **New-CMUserCollection** cmdlet creates a collection based on a specific limiting collection. The limiting collection determines which users can be a member of the user collection that you create. For example, when you use the All Users collection as the limiting collection, the new collection can include any user in the Microsoft System Center 2012 Configuration Manager hierarchy. You specify the limiting collection by providing its name or ID.

Users are added to the collection by membership rules. To add members to the user collection use one of the following membership rule cmdlets:

- [Add-CMUserCollectionDirectMembershipRule](#)
- [Add-CMUserCollectionExcludeMembershipRule](#)
- [Add-CMUserCollectionIncludeMembershipRule](#)
- [Add-CMDeviceCollectionQueryMembershipRule](#)

Collections represent logical groupings of resources, such as users and devices. For more information about Configuration Manager collections, see [Introduction to Collections in Configuration Manager](http://go.microsoft.com/fwlink/p/?LinkID=259433) (<http://go.microsoft.com/fwlink/p/?LinkID=259433>) on TechNet.

Parameters

-Comment<String>

Specifies a description of the collection, such as the users in a specific organization.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-LimitingCollectionId<String>

Specifies the ID of the limiting collection that Configuration Manager uses to limit which users are available to the user collection that you are creating. For example, the following ID is the ID of the All Users collection: SMS00002.

Aliases	LimitToCollectionId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-LimitingCollectionName<String>

Specifies the name of the limiting collection that Configuration Manager uses to limit which users are available to the user collection that you are creating. For example, you can specify the All Users collection.

Aliases	LimitToCollectionName
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies the name for the user collection that you want to create.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RefreshSchedule<IResultObject>

Specifies a schedule that determines when Configuration Manager refreshes the collection.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RefreshType<RefreshTypes>

Specifies how Configuration Manager refreshes the collection. Valid values are:

-- Manual. The collection is refreshed using the Configuration Manager console or the Configuration Manager SDK.

-- Periodic. The collection is refreshed based on the schedule specified by the *RefreshSchedule* parameter.

-- ConstantUpdate. The collection is refreshed whenever a member is added to the collection.

The acceptable values for this parameter are:

Both	
ConstantUpdate	
Manual	

Periodic	
----------	--

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Create a user collection

This command creates a collection for all users in the Sales department. Specifying All Users for the *LimitingCollectionName* parameter indicates that the new collection can include any user in the Configuration Manager hierarchy.

```
PS C:\> New-CMUserCollection -Name "Sales" -LimitingCollectionName "All Users"
```

Related topics

[Get-CMUserCollection](#)

[Set-CMUserCollection](#)

[Remove-CMUserCollection](#)

[Import-CMUserCollection](#)

[Export-CMUserCollection](#)

New-CMUserDataAndProfileConfigurationItem

New-CMUserDataAndProfileConfigurationItem

Creates a user data and profile configuration item.

Syntax

Parameter Set: New

```
New-CMUserDataAndProfileConfigurationItem -ConfigureFolderRedirection <Boolean> -
ConfigureOfflineFile <Boolean> -ConfigureRoamingUserProfile <Boolean> -Name <String> [-
AccessPolicy <Boolean> ] [-AddAdminGroupToRUPEnabled <Boolean> ] [-AllowAllDevice <Boolean> ]
[-AllowCrossForestUserPolicy <Boolean> ] [-BackgroundSynchronization {Disabled | Enabled |
NotConfigured} ] [-ConnectionTransferRate <Int32> ] [-DeleteProfileOlderDays <Int32> ] [-
DeleteRoamingCacheEnabled <Boolean> ] [-Description <String> ] [-DetectSlowLinkDisabled
<Boolean> ] [-DeviceType {FolderRedirectionOnAnyDeviceCachingOnPrimaryDevicesOnly |
OnAnyDevice | OnlyOnPrimaryDevices} ] [-DisableMakeOffline <Boolean> ] [-DisableWorkOffline
<Boolean> ] [-EnableOfflineFile <Boolean> ] [-EnableSlowLink <Boolean> ] [-ErrorDays <Int32>
] [-Excludelist <String[]> ] [-FileSynchronization {Disabled | Enabled | NotConfigured} ] [-
FolderRedirectionUserConfigurationForAppDataRoaming {DoNotManage | RedirectToLocal |
RedirectToRemote} ] [-FolderRedirectionUserConfigurationForContacts {DoNotManage |
RedirectToLocal | RedirectToRemote} ] [-FolderRedirectionUserConfigurationForDesktop
{DoNotManage | RedirectToLocal | RedirectToRemote} ] [-
FolderRedirectionUserConfigurationForDocuments {DoNotManage | RedirectToLocal |
RedirectToRemote} ] [-FolderRedirectionUserConfigurationForDownloads {DoNotManage |
RedirectToLocal | RedirectToRemote} ] [-FolderRedirectionUserConfigurationForFavorites
{DoNotManage | RedirectToLocal | RedirectToRemote} ] [-
FolderRedirectionUserConfigurationForLinks {DoNotManage | RedirectToLocal |
RedirectToRemote} ] [-FolderRedirectionUserConfigurationForMusic {DoNotManage |
RedirectToLocal | RedirectToRemote} ] [-FolderRedirectionUserConfigurationForPictures
{DoNotManage | RedirectToLocal | RedirectToRemote} ] [-
FolderRedirectionUserConfigurationForSavedGames {DoNotManage | RedirectToLocal |
RedirectToRemote} ] [-FolderRedirectionUserConfigurationForSearches {DoNotManage |
RedirectToLocal | RedirectToRemote} ] [-FolderRedirectionUserConfigurationForStartMenu
{DoNotManage | RedirectToLocal | RedirectToRemote} ] [-
FolderRedirectionUserConfigurationForVideos {DoNotManage | RedirectToLocal |
RedirectToRemote} ] [-ForceUnloadDisabled <Boolean> ] [-GrantExclusiveRight <Boolean> ] [-
LeaveFolderNewLocation <Boolean> ] [-LimitDisk <Int32> ] [-ManageAdvancedSetting <Boolean> ]
[-ManageSlowLink <Boolean> ] [-MoveCachedFolder <Boolean> ] [-MoveContent <Boolean> ] [-
OfflineFile <String[]> ] [-OnlyAllowLocalProfiles <Boolean> ] [-OwnerCheckDisabled <Boolean>
] [-ProfileUploadDisabled <Boolean> ] [-SlowLink <Int32> ] [-SlowLinkUIEnabled <Boolean> ]
[-SpecifiedLocation <String> ] [-SpecifyTime <String> ] [-SpecifyTimeInterval <Int32> ] [-
SynchronizationInterval <Int32> ] [-SynchronizationList <String[]> ] [-SynchronizationPolicy
<Boolean> ] [-TempProfileLogonBlocked <Boolean> ] [-TimeOut <Int32> ] [-UseCommonAlert
<Boolean> ] [-UseSpecifiedLocation <Boolean> ] [-WaitForNetworkInSeconds <Int32> ] [-
WarningDays <Int32> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **New-CMUserDataAndProfileConfigurationItem** cmdlet creates a user data and profile configuration item that can apply to Windows 8 computers. A configuration item can manage folder redirection, offline folders, and roaming user profiles. You can modify settings for an existing configuration item by using the **Set-CMUserDataAndProfileConfigurationItem** cmdlet.

Parameters

-AccessPolicy<Boolean>

Indicates whether this configuration item manages profile access settings for roaming profiles.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AddAdminGroupToRUPEnabled<Boolean>

Indicates whether to grant the Administrators group access to roaming profile folders.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AllowAllDevice<Boolean>

Indicates whether to allow roaming profiles on all devices. If this value is `$False`, roaming profiles apply only to the primary device for a user.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AllowCrossForestUserPolicy<Boolean>

Indicates whether to permit user policies to roam across Active Directory forests that have a trust relationship with the current forest.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-BackgroundSynchronization<SynchronizationType>

Specifies a background synchronization type for file in offline mode. Valid values are: Disabled, Enabled, and NotConfigured.

The acceptable values for this parameter are:

Disabled	
Enabled	
NotConfigured	

Aliases	none
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ConfigureFolderRedirection<Boolean>

Indicates whether the configuration item includes settings for folder redirection.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ConfigureOfflineFile<Boolean>

Indicates whether the configuration item includes settings for offline folders.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ConfigureRoamingUserProfile<Boolean>

Indicates whether the configuration item includes settings for roaming user profiles.

Aliases	none
Required?	true

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ConnectionTransferRate<Int32>

Specifies a connection transfer rate.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeleteProfileOlderDays<Int32>

Specifies the number of days to keep a user profile since the last time someone used it. A computer deletes an older profile when it restarts.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeleteRoamingCacheEnabled<Boolean>

Indicates whether to delete cached copies of roaming user profiles. The default for this parameter is \$False.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Description<String>

Specifies a description for the configuration item.

Aliases	LocalizedDescription
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DetectSlowLinkDisabled<Boolean>

Indicates whether to disable slow link detection.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeviceType<DeviceType>

Specifies the applicability of folder redirection for user devices. Valid values are:

-- FolderRedirectionOnAnyDeviceCachingOnPrimaryDevicesOnly. Folder redirection for any user device, but caching only on the primary device for a user.

-- OnAnyDevice. Folder redirection and caching on any device.

-- OnlyOnPrimaryDevices. Folder redirection and caching on the primary device for a user.

The acceptable values for this parameter are:

FolderRedirectionOnAnyDeviceCachingOnPrimaryDevicesOnly	
OnAnyDevice	
OnlyOnPrimaryDevices	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DisableMakeOffline<Boolean>

Indicates whether users can disable the **Make Available Offline** command.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DisableWorkOffline<Boolean>

Indicates whether users can disable the **Work Offline** command.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnableOfflineFile<Boolean>

Indicates whether this configuration item enables use of offline files.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnableSlowLink<Boolean>

Indicates whether the configuration enables work with offline files over a slow link.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ErrorDays<Int32>

Specifies the number of days to wait before the profile creates an error.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ExcludeList<String[]>

Specifies an array of folders. The configuration item excludes these folders from roaming profiles.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-FileSynchronization<SynchronizationType>

Specifies a file synchronization type for metered networks for work in offline mode. Valid values are: Disabled, Enabled, and NotConfigured.

The acceptable values for this parameter are:

Disabled	
Enabled	
NotConfigured	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-

FolderRedirectionUserConfigurationForAppDataRoaming<FolderRedirectionType>

Specifies whether to redirect the Application Data folder to a local folder or a remote folder, or not to redirect this folder. Valid values are:

- DoNotManage
- RedirectToLocal
- RedirectToRemote

The acceptable values for this parameter are:

DoNotManage	
RedirectToLocal	
RedirectToRemote	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-

FolderRedirectionUserConfigurationForContacts<FolderRedirectionType>

Specifies whether to redirect the Contacts folder to a local folder or a remote folder, or not to redirect this folder. Valid values are:

- DoNotManage
- RedirectToLocal
- RedirectToRemote

The acceptable values for this parameter are:

DoNotManage	
RedirectToLocal	
RedirectToRemote	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-

FolderRedirectionUserConfigurationForDesktop<FolderRedirectionType>

Specifies whether to redirect the Desktop to a local folder or a remote folder, or not to redirect this folder. Valid values are:

- DoNotManage
- RedirectToLocal
- RedirectToRemote

The acceptable values for this parameter are:

DoNotManage	
RedirectToLocal	
RedirectToRemote	

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-

FolderRedirectionUserConfigurationForDocuments<FolderRedirectionType>

Specifies whether to redirect the Documents folder to a local folder or a remote folder, or not to redirect this folder. Valid values are:

- DoNotManage
- RedirectToLocal
- RedirectToRemote

The acceptable values for this parameter are:

DoNotManage	
RedirectToLocal	
RedirectToRemote	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-

FolderRedirectionUserConfigurationForDownloads<FolderRedirectionType>

Specifies whether to redirect the Downloads folder to a local folder or a remote folder, or not to redirect this folder. Valid values are:

- DoNotManage
- RedirectToLocal
- RedirectToRemote

The acceptable values for this parameter are:

DoNotManage	
RedirectToLocal	
RedirectToRemote	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-

FolderRedirectionUserConfigurationForFavorites<FolderRedirection Type>

Specifies whether to redirect the Favorites folder to a local folder or a remote folder, or not to redirect this folder. Valid values are:

- DoNotManage
- RedirectToLocal
- RedirectToRemote

The acceptable values for this parameter are:

DoNotManage	
RedirectToLocal	
RedirectToRemote	

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-

FolderRedirectionUserConfigurationForLinks<FolderRedirectionType>

Specifies whether to redirect the Links folder to a local folder or a remote folder, or not to redirect this folder. Valid values are:

- DoNotManage
- RedirectToLocal
- RedirectToRemote

The acceptable values for this parameter are:

DoNotManage	
RedirectToLocal	
RedirectToRemote	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-

FolderRedirectionUserConfigurationForMusic<FolderRedirectionType>

Specifies whether to redirect the Music folder to a local folder or a remote folder, or not to redirect this folder. Valid values are:

- DoNotManage
- RedirectToLocal
- RedirectToRemote

The acceptable values for this parameter are:

DoNotManage	
RedirectToLocal	
RedirectToRemote	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-

FolderRedirectionUserConfigurationForPictures<FolderRedirectionType>

Specifies whether to redirect the Pictures folder to a local folder or a remote folder, or not to redirect this folder. Valid values are:

- DoNotManage
- RedirectToLocal
- RedirectToRemote

The acceptable values for this parameter are:

DoNotManage	
RedirectToLocal	
RedirectToRemote	

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-

FolderRedirectionUserConfigurationForSavedGames<FolderRedirectionType>

Specifies whether to redirect the Saved Games folder to a local folder or a remote folder, or not to redirect this folder. Valid values are:

- DoNotManage
- RedirectToLocal
- RedirectToRemote

The acceptable values for this parameter are:

DoNotManage	
RedirectToLocal	
RedirectToRemote	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-

FolderRedirectionUserConfigurationForSearches<FolderRedirectionType>

Specifies whether to redirect the Searches folder to a local folder or a remote folder, or not to redirect this folder. Valid values are:

- DoNotManage
- RedirectToLocal
- RedirectToRemote

The acceptable values for this parameter are:

DoNotManage	
RedirectToLocal	
RedirectToRemote	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-

FolderRedirectionUserConfigurationForStartMenu<FolderRedirectionType>

Specifies whether to redirect the Start Menu to a local folder or a remote folder, or not to redirect this folder. Valid values are:

- DoNotManage
- RedirectToLocal
- RedirectToRemote

The acceptable values for this parameter are:

DoNotManage	
RedirectToLocal	
RedirectToRemote	

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-

FolderRedirectionUserConfigurationForVideos<FolderRedirectionType>

Specifies whether to redirect the Videos folder to a local folder or a remote folder, or not to redirect this folder. Valid values are:

- DoNotManage
- RedirectToLocal
- RedirectToRemote

The acceptable values for this parameter are:

DoNotManage	
RedirectToLocal	
RedirectToRemote	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ForceUnloadDisabled<Boolean>

Indicates whether to disable forced unload of a user profile at logoff. The default value for this parameter is \$False.

Aliases	none
Required?	false
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-GrantExclusiveRight<Boolean>

Indicates whether to grant the user exclusive permissions to a redirected folder.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-LeaveFolderNewLocation<Boolean>

Indicates whether to leave the folder in the redirected location in the event you remove this configuration item.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-LimitDisk<Int32>

Specifies a limit, in megabytes, for the disk space used for offline files.

Aliases	none
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ManageAdvancedSetting<Boolean>

Indicates whether this configuration item manages advanced settings for folder redirection. Specify values for any of the following parameters:

- *GrantExclusiveRight*
- *MoveContent*
- *LeaveFolderNewLocation*
- *MoveCachedFolder*

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ManageSlowLink<Boolean>

Indicates whether this profile item manages slow links.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-MoveCachedFolder<Boolean>

Indicates whether to move the cached folder when the path updates on the server.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-MoveContent<Boolean>

Indicates whether to move the contents of redirected folders to the new location.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies a name for the configuration item.

Aliases	LocalizedDisplayName
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OfflineFile<String[]>

Specifies an array of Administrative user assigned offline folders, as UNC paths as follows:
\\serve\share\%UserName%.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OnlyAllowLocalProfiles<Boolean>

Indicates whether the configuration item allows only local profiles, not domain profiles.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OwnerCheckDisabled<Boolean>

Indicates whether the configuration item does not check for ownership of roaming profile folders.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ProfileUploadDisabled<Boolean>

Indicates whether to disable uploading of profiles.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SlowLink<Int32>

Specifies a value for a slow link.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SlowLinkUIEnabled<Boolean>

Indicates whether to enable user logon prompt to allow profile download when a device detects a slow link.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SpecifiedLocation<String>

Specifies a specified location.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SpecifyTime<String>

Specifies a time for background upload of the user hive.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SpecifyTimeInterval<Int32>

Specifies a time of day for synchronization for user profiles. Enter a value of zero (0) to 12. Zero indicates 12:00 a.m. Values from one (1) to 12 indicate times from 1:00 p.m. to 12:00 p.m.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SynchronizationInterval<Int32>

Specifies a synchronization interval, in hours, for the user profiles.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SynchronizationList<String[]>

Specifies an array of folders. The configuration item specifies these subfolders of Appdata\Roaming to synchronize only at logon and logoff.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SynchronizationPolicy<Boolean>

Indicates whether to use a synchronization policy.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-TempProfileLogonBlocked<Boolean>

Indicates whether to block users from logging on by using a temporary profile.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-TimeOut<Int32>

Specifies a timeout value.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UseCommonAlert<Boolean>

Indicates whether to use common alerts.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UseSpecifiedLocation<Boolean>

Indicates whether to use the specified location referred to by the *SpecifiedLocation* parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-WaitForNetworkInSeconds<Int32>

Specifies the maximum time to wait, in seconds, for slow link network connectivity before loading the profile.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-WarningDays<Int32>

Specifies the number of days to wait before the profile creates a warning.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Specify folder redirection

This command creates a configuration item named CMUserDataProfileCI07. This configuration item manages folder redirection, offline folders, and roaming user profiles. The command specifies redirect to remote for the Desktop folder. The command includes a specified location and includes the *UseSpecifiedLocation* parameter with a value of \$True.

```
PS C:\> New-CMUserDataAndProfileConfigurationItem -ConfigureFolderRedirection $True -  
ConfigureOfflineFile $True -ConfigureRoamingUserProfile $True -Name "CMUserDataProfileCI07" -  
FolderRedirectionUserConfigurationForDesktop RedirectToRemote -SpecifiedLocation  
"\\Server017\Shared" -UseSpecifiedLocation $True
```

Example 2: Specify settings for a configuration item

This command creates a configuration item named CMUserDataProfileCI27. This configuration item manages folder redirection, offline folders, and roaming user profiles. Like the previous example, this command specifies folder redirection settings, this time for several folders. The command includes other settings for folder redirection, offline folders, and roaming user profiles.

```
PS C:\> New-CMUserDataAndProfileConfigurationItem -ConfigureFolderRedirection $True -
ConfigureOfflineFile $True -ConfigureRoamingUserProfile $True -Name "CMUserDataProfileCI27" -
AccessPolicy $True -AddAdminGroupToRUPEnabled $True -AllowAllDevice $False -
AllowCrossForestUserPolicy $True -BackgroundSynchronization NotConfigured -
ConnectionTransferRate 300 -DeleteProfileOlderDays 300 -DeleteRoamingCacheEnabled $True -
Description "Configuration item for off-site staff." -DetectSlowLinkDisabled $False -
DeviceType OnlyOnPrimaryDevices -DisableMakeOffline $True -DisableWorkOffline $True -
EnableOfflineFile $True -EnableSlowLink $True -ErrorDays 200 -ExcludeList "MiscFolder" -
FileSynchronization Enabled -FolderRedirectionUserConfigurationForAppDataRoaming
RedirectToLocal -FolderRedirectionUserConfigurationForContacts RedirectToRemote -
FolderRedirectionUserConfigurationForDesktop RedirectToRemote -
FolderRedirectionUserConfigurationForDocuments RedirectToRemote -
FolderRedirectionUserConfigurationForDownloads RedirectToLocal -
FolderRedirectionUserConfigurationForFavorites RedirectToLocal -
FolderRedirectionUserConfigurationForLinks RedirectToRemote -
FolderRedirectionUserConfigurationForMusic RedirectToLocal -
FolderRedirectionUserConfigurationForPictures RedirectToRemote -
FolderRedirectionUserConfigurationForSavedGames RedirectToRemote -
FolderRedirectionUserConfigurationForSearches RedirectToLocal -
FolderRedirectionUserConfigurationForStartMenu RedirectToLocal -
FolderRedirectionUserConfigurationForVideos RedirectToRemote -ForceUnloadDisabled $True -
GrantExclusiveRight $False -LeaveFolderNewLocation $True -LimitDisk 1024 -
ManageAdvancedSetting $True -ManageSlowLink $True -MoveCachedFolder $True -MoveContent
$False -OfflineFile "\\Server78\MiscShared" -OnlyAllowLocalProfiles $True -
OwnerCheckDisabled $True -ProfileUploadDisabled $True -SlowLink 350 -SlowLinkUIEnabled $True
-SpecifiedLocation "\\Server221\Part17" -SpecifyTimeInterval 120 -SynchronizationInterval
1200 -SynchronizationList "\\Server221\Part17" -SynchronizationPolicy $True -
TempProfileLogonBlocked $True -TimeOut 130 -UseCommonAlert $True -UseSpecifiedLocation $True
-WaitForNetworkInSeconds 300 -WarningDays 100
```

Related topics

[Copy-CMUserDataAndProfileConfigurationItem](#)

[Get-CMUserDataAndProfileConfigurationItem](#)

[Remove-CMUserDataAndProfileConfigurationItem](#)

[Set-CMUserDataAndProfileConfigurationItem](#)

New-CMVhd

New-CMVhd

Creates a VHD image.

Syntax

Parameter Set: New

```
New-CMVhd -DistributionPointServerNames <String[]> -Name <String> -Path <String> -  
TaskSequencePackageId <String> [-Description <String> ] [-Version <String> ] [-VHDSize  
<Int32> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **New-CMVhd** cmdlet creates a virtual hard disk (VHD) image by using the operating system deployment feature.

Parameters

-Description<String>

Specifies a description for the VHD.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DistributionPointServerNames<String[]>

Specifies an array of names of distribution point servers, which contain content that the task sequence requires.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies the name of a VHD image.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Path<String>

Specifies a path for the VHD image.

Aliases	PackageSourcePath
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-TaskSequencePackageId<String>

Specifies an ID for a task sequence package.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Version<String>

Specifies a version for the VHD. Use any string.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-VHDSize<Int32>

Specifies the initial size, in gigabytes, of a dynamically expanding VHD. The default value is 50 GB.

Aliases	none
Required?	false
Position?	named
Default Value	50 GB
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](#).

Related topics

[Get-CMVhd](#)

[Remove-CMVhd](#)

[Set-CMVhd](#)

New-CMVirtualEnvironmentGroup

New-CMVirtualEnvironmentGroup

Creates a virtual environment group.

Syntax

Parameter Set: New

```
New-CMVirtualEnvironmentGroup -DeploymentType <DeploymentTypeItem[]> -Name <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **New-CMVirtualEnvironmentGroup** cmdlet creates a virtual environment group that specifies deployment type items.

A virtual environment allows two or more Microsoft Application Virtualization (App-V) deployment types to share the same file system and registry on client computers. When multiple virtual applications modify the same file system or registry values on a client computer, the application with the highest order takes precedence. When an application is installed or when a client evaluates installed applications, the virtual environment is added or modified on client computers.

Parameters

-DeploymentType<DeploymentTypeItem[]>

Specifies an array of deployment type items.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies a name for the virtual environment group.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Create a virtual environment group

This command creates a virtual environment group named Office Remote Apps for the deployment type Office_Standard.

```
PS C:\> New-CMVirtualEnvironmentGroup -DeploymentType "Office_Standard" -Name "Office Remote Apps"
```

New-CMVpnProfileConfigurationItem

New-CMVpnProfileConfigurationItem

Creates a VPN profile.

Syntax

Parameter Set: Default

```
New-CMVpnProfileConfigurationItem -DesiredConfigurationDigestPath <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **New-CMVpnProfileConfigurationItem** cmdlet creates a virtual private network (VPN) profile. Client computers use VPN profiles to remotely connect to a company network over the Internet.

Parameters

-DesiredConfigurationDigestPath<String>

Specifies a path to the configuration data stored as a digest.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

[Copy-CMVpnProfileConfigurationItem](#)

[Get-CMVpnProfileConfigurationItem](#)

[Remove-CMVpnProfileConfigurationItem](#)

[Set-CMVpnProfileConfigurationItem](#)

New-CMWindowsFirewallPolicy

New-CMWindowsFirewallPolicy

Creates a new Windows Firewall policy in Configuration Manager.

Syntax

Parameter Set: New

```
New-CMWindowsFirewallPolicy -Name <String> [-Description <String> ] [-DomainBlockAllInboundTraffic {No | NotConfigured | Yes} ] [-DomainNotifications {No | NotConfigured | Yes} ] [-DomainTurnOnFirewall {No | NotConfigured | Yes} ] [-PrivateBlockAllInboundTraffic {No | NotConfigured | Yes} ] [-PrivateNotifications {No | NotConfigured | Yes} ] [-PrivateTurnOnFirewall {No | NotConfigured | Yes} ] [-PublicBlockAllInboundTraffic {No | NotConfigured | Yes} ] [-PublicNotifications {No | NotConfigured | Yes} ] [-PublicTurnOnFirewall {No | NotConfigured | Yes} ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **New-CMWindowsFirewallPolicy** cmdlet creates a configuration policy for Windows Firewall in Microsoft System Center 2012 Configuration Manager.

Windows Firewall allows or denies incoming connections to an IP address. The blocking actions allow or deny incoming traffic based on a network location type. The network location types are: domain, public, and private.

Parameters

-Description<String>

Specifies a description for the firewall policy.

Aliases	LocalizedDescription
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-DomainBlockAllInboundTraffic<SettingType>

Specifies whether to block all incoming traffic for a domain type of network location. Valid values are:

- No
- Not Configured
- Yes

The acceptable values for this parameter are:

No	
NotConfigured	
Yes	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DomainNotifications<SettingType>

Specifies whether to enable notifications for a domain type of network location. Valid values are:

- No
- Not Configured
- Yes

The acceptable values for this parameter are:

No	
NotConfigured	
Yes	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DomainTurnOnFirewall<SettingType>

Specifies whether to turn on a firewall for a domain type of network location. Valid values are:

- No
- Not Configured
- Yes

The acceptable values for this parameter are:

No	
NotConfigured	
Yes	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies a name for the firewall policy in Configuration Manager.

Aliases	LocalizedDisplayName
---------	----------------------

Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PrivateBlockAllInboundTraffic<SettingType>

Specifies whether to block all incoming traffic for a private type of network location. Valid values are:

- No
- Not Configured
- Yes

The acceptable values for this parameter are:

No	
NotConfigured	
Yes	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PrivateNotifications<SettingType>

Specifies whether to enable notifications for a private type of network location. Valid values are:

- No
- Not Configured
- Yes

The acceptable values for this parameter are:

No	
NotConfigured	
Yes	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PrivateTurnOnFirewall<SettingType>

Specifies whether to turn on a firewall for a private type of network location. Valid values are:

- No
- Not Configured
- Yes

The acceptable values for this parameter are:

No	
NotConfigured	
Yes	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PublicBlockAllInboundTraffic<SettingType>

Specifies whether to block all incoming traffic for a public type of network location. Valid values are:

- No
- Not Configured
- Yes

The acceptable values for this parameter are:

No	
NotConfigured	
Yes	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PublicNotifications<SettingType>

Specifies whether Configuration Manager sends notifications to public network locations. Valid values are:

- No
- Not Configured
- Yes

The acceptable values for this parameter are:

No	
NotConfigured	
Yes	

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PublicTurnOnFirewall<SettingType>

Specifies whether to enable Windows Firewall for a public network location. Valid values are:

- No
- Not Configured
- Yes

The acceptable values for this parameter are:

No	
NotConfigured	
Yes	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Create a Windows Firewall policy

This command creates a new Windows Firewall policy and enables the firewall for domain, private, and public network location types.

```
PS C:\> New-CMWindowsFirewallPolicy -Name "test01" -Description "323132" -  
DomainTurnOnFirewall Yes -PrivateTurnOnFirewall Yes -PublicTurnOnFirewall Yes
```

Related topics

[Get-CMWindowsFirewallPolicy](#)

[Remove-CMWindowsFirewallPolicy](#)

[Set-CMWindowsFirewallPolicy](#)

New-CMWiredProfileObject

New-CMWiredProfileObject

Creates a profile that specifies settings for AMT-based computers on a wired network.

Syntax

Parameter Set: NewWiredProfileObject

```
New-CMWiredProfileObject -ClientAuthenticationMethod {EapTls | EapTtlsMschapv2 |  
Peapv0EapMschapv2} -ClientCertificateTemplate <String> -ClientCertificationAuthorityName  
<String> -ClientIssuingCertificationAuthority <String> -TrustedRootCertificate  
<X509Certificate2> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **New-CMWiredProfileObject** cmdlet creates a Microsoft System Center 2012 Configuration Manager profile that specifies settings that Intel Active Management Technology (Intel AMT)-based computers use on a wired network. These settings must match the configuration on your Remote Authentication Dial-In User Service (RADIUS) server. System Center 2012 Configuration Manager cannot validate that these settings with your RADIUS server.

Parameters

-ClientAuthenticationMethod<ClientAuthenticationMethodType>

Specifies the client authentication method configured on your RADIUS server. Valid values are:

- EapTls. EAP-TLS.
- EapTtlsMschapv2. EAP-TTLS/MSCHAPv2.
- Peapv0EapMschapv2. PEAPv0/EAP-MSCHAPv2.

The default authentication method is EAP-TLS.

The acceptable values for this parameter are:

EapTls	
EapTtlsMschapv2	
Peapv0EapMschapv2	

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ClientCertificateTemplate<String>

Specifies a client certificate template.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ClientCertificationAuthorityName<String>

Specifies a certification authority for the client.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ClientIssuingCertificationAuthority<String>

Specifies an issuing certification authority for the client.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-TrustedRootCertificate<X509Certificate2>

Specifies the trusted root certificate that the RADIUS server uses as its server authentication certificate.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Create a profile for AMT-based computers on a wired network

This command creates a profile for Intel AMT-based computers on a wired network. The command specifies security settings, such as the client authentication method and information necessary for certificates. These settings must match the settings for the RADIUS server.

```
PS C:\> New-CMWiredProfileObject -ClientAuthenticationMethod EapTtlsMschapv2 -
ClientCertificateTemplate "Contoso Wired User" -ClientCertificationAuthorityName "Contoso CA
1" -ClientIssuingCertificationAuthority "ContosoCA.Contoso.com" -TrustedRootCertificate
"Contoso Root"
```

Related topics

[New-CMWirelessProfileObject](#)

New-CMWirelessProfileConfigurationItem

New-CMWirelessProfileConfigurationItem

Creates a wireless profile.

Syntax

Parameter Set: Default

```
New-CMWirelessProfileConfigurationItem -DesiredConfigurationDigestPath <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **New-CMWirelessProfileConfigurationItem** cmdlet creates a wireless profile. Client computers use wireless profiles for configuration when they connect to a company wireless network.

Parameters

-DesiredConfigurationDigestPath<String>

Specifies a path to the configuration data stored as a digest.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

[Copy-CMWirelessProfileConfigurationItem](#)

[Get-CMWirelessProfileConfigurationItem](#)

[Remove-CMWirelessProfileConfigurationItem](#)

[Set-CMWirelessProfileConfigurationItem](#)

New-CMWirelessProfileObject

New-CMWirelessProfileObject

Creates a profile that specifies settings for AMT-based computers on a wireless network.

Syntax

Parameter Set: NewWirelessProfileObject

```
New-CMWirelessProfileObject -ClientAuthenticationMethod {EapTls | EapTtlsMschapv2 | Peapv0EapMschapv2} -ClientCertificateTemplate <String> -ClientCertificationAuthorityName <String> -ClientIssuingCertificationAuthority <String> -EncryptionMethod {AES | TKIP} -NetworkName <String> -ProfileName <String> -SecurityType {WPA2Enterprise | WPAEnterprise} -TrustedRootCertificate <X509Certificate2> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **New-CMWirelessProfileObject** cmdlet creates a Microsoft System Center 2012 Configuration Manager profile that specifies settings that Intel Active Management Technology (Intel AMT)-based computers use on a wireless network. These settings must match the configuration on your Remote Authentication Dial-In User Service (RADIUS) server. System Center 2012 Configuration Manager cannot validate that these settings with your RADIUS server.

Parameters

-ClientAuthenticationMethod<ClientAuthenticationMethodType>

Specifies the client authentication method configured on your RADIUS server. Valid values are:

- EapTls. EAP-TLS.
- EapTtlsMschapv2. EAP-TTLS/MSCHAPv2.
- Peapv0EapMschapv2. PEAPv0/EAP-MSCHAPv2.

The default authentication method is EAP-TLS.

The acceptable values for this parameter are:

EapTls	
EapTtlsMschapv2	
Peapv0EapMschapv2	

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ClientCertificateTemplate<String>

Specifies a client certificate template.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ClientCertificationAuthorityName<String>

Specifies a certification authority for the client.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ClientIssuingCertificationAuthority<String>

Specifies an issuing certification authority for the client.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EncryptionMethod<EncryptionType>

Specifies the encryption method. This method is the same as the encryption method for your wireless network. Valid values are: AES and TKIP. The default value for this parameter is AES.

The acceptable values for this parameter are:

AES	
TKIP	

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-NetworkName<String>

Specifies the name of your wireless network, also known as the service set identifier (SSID). This name cannot exceed 32 characters.

Aliases	none
Required?	true
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ProfileName<String>

Specifies a name for this profile. Use alphanumeric characters only, start the name with a letter, and do not exceed 32 characters.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SecurityType<WirelessProfileSecurityType>

Specifies the security type for your wireless network. Valid values are: WPA2Enterprise and WPAEnterprise. The default value for this parameter is WPA2Enterprise.

The acceptable values for this parameter are:

WPA2Enterprise	
WPAEnterprise	

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-TrustedRootCertificate<X509Certificate2>

Specifies the trusted root certificate that the RADIUS server uses as its server authentication certificate.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Create a profile for AMT-based computers on a wireless network

This command creates a profile for Intel AMT-based computers on a wireless network. The command specifies security settings, such as the client authentication method, information necessary for certificates, and wireless network configuration. These settings must match the settings for the RADIUS server and wireless network.

```
PS C:\> New-CMWirelessProfileObject -ClientAuthenticationMethod EapTlsMschapv2 -
ClientCertificateTemplate "Contoso Wireless User" -ClientCertificationAuthorityName
"Contoso CA 1" -ClientIssuingCertificationAuthority "ContosoCA.Contoso.com" -
EncryptionMethod TKIP -NetworkName "ContosoWireless" -ProfileName "WirelessLocalUser" -
SecurityType WPA2Enterprise -TrustedRootCertificate "Contoso Root CA"
```

Related topics

[New-CMWiredProfileObject](#)

Publish-CMPrestageContent

Publish-CMPrestageContent

Publishes files to a distribution point.

Syntax

Parameter Set: SearchByIdMandatory_Application

```
Publish-CMPrestageContent -ApplicationId <String[]> -DistributionPointName <String> -  
FileName <String> [-Description <String> ] [-DisableExportAllDependencies] [-Confirm] [-  
WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory_BootImage

```
Publish-CMPrestageContent -BootImageId <String[]> -DistributionPointName <String> -FileName  
<String> [-Description <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory_DeploymentPackage

```
Publish-CMPrestageContent -DeploymentPackageId <String[]> -DistributionPointName <String> -  
FileName <String> [-Description <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory_DriverPackage

```
Publish-CMPrestageContent -DistributionPointName <String> -DriverPackageId <String[]> -  
FileName <String> [-Description <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory_OperatingSystemImage

```
Publish-CMPrestageContent -DistributionPointName <String> -FileName <String> -  
OperatingSystemImageId <String[]> [-Description <String> ] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByIdMandatory_OperatingSystemInstaller

```
Publish-CMPrestageContent -DistributionPointName <String> -FileName <String> -  
OperatingSystemInstallerId <String[]> [-Description <String> ] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByIdMandatory_Package

```
Publish-CMPrestageContent -DistributionPointName <String> -FileName <String> -PackageId  
<String[]> [-Description <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByNameMandatory_Application

```
Publish-CMPrestageContent -ApplicationName <String[]> -DistributionPointName <String> -  
FileName <String> [-Description <String> ] [-DisableExportAllDependencies] [-Confirm] [-  
WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByNameMandatory_BootImage

```
Publish-CMPrestageContent -BootImageName <String[]> -DistributionPointName <String> -  
FileName <String> [-Description <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByNameMandatory_DeploymentPackage

Publish-CMPrestageContent -DeploymentPackageName <String[]> -DistributionPointName <String> -FileName <String> [-Description <String>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SearchByNameMandatory_DriverPackage

Publish-CMPrestageContent -DistributionPointName <String> -DriverPackageName <String[]> -FileName <String> [-Description <String>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SearchByNameMandatory_OperatingSystemImage

Publish-CMPrestageContent -DistributionPointName <String> -FileName <String> -OperatingSystemImageName <String[]> [-Description <String>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SearchByNameMandatory_OperatingSystemInstaller

Publish-CMPrestageContent -DistributionPointName <String> -FileName <String> -OperatingSystemInstallerName <String[]> [-Description <String>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SearchByNameMandatory_Package

Publish-CMPrestageContent -DistributionPointName <String> -FileName <String> -PackageName <String[]> [-Description <String>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SearchByValueMandatory_Application

Publish-CMPrestageContent -Application <IResultObject> -DistributionPointName <String> -FileName <String> [-Description <String>] [-DisableExportAllDependencies] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SearchByValueMandatory_BootImage

Publish-CMPrestageContent -BootImage <IResultObject> -DistributionPointName <String> -FileName <String> [-Description <String>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SearchByValueMandatory_DeploymentPackage

Publish-CMPrestageContent -DeploymentPackage <IResultObject> -DistributionPointName <String> -FileName <String> [-Description <String>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SearchByValueMandatory_DriverPackage

Publish-CMPrestageContent -DistributionPointName <String> -DriverPackage <IResultObject> -FileName <String> [-Description <String>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SearchByValueMandatory_OperatingSystemImage

Publish-CMPrestageContent -DistributionPointName <String> -FileName <String> -OperatingSystemImage <IResultObject> [-Description <String>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SearchByValueMandatory_OperatingSystemInstaller

Publish-CMPrestageContent -DistributionPointName <String> -FileName <String> -OperatingSystemInstaller <IResultObject> [-Description <String>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SearchByValueMandatory_Package

Publish-CMPrestageContent -DistributionPointName <String> -FileName <String> -Package <IResultObject> [-Description <String>] [-Confirm] [-WhatIf] [<CommonParameters>]

Detailed Description

The **Publish-CMPrestageContent** cmdlet publishes files for applications, images, packages, or operating system installers to a distribution point without using the Microsoft System Center 2012 Configuration Manager distribution process. Specify the distribution site, the file name, and the item to publish.

You can specify any of the following to publish to a distribution point:

- Application
- BootImage
- DeploymentPackage
- DriverPackage
- OperatingSystemImage
- OperatingSystemInstaller
- Package

You can specify the item to be published by name or ID, or use another cmdlet to get the desired item.

Parameters

-Application<IResultObject>

Specifies an application object. To obtain an application object, use the **Get-CMApplication** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ApplicationId<String[]>

Specifies an array of IDs of applications.

Aliases	none
---------	------

Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ApplicationName<String[]>

Specifies an array of names of applications.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-BootImage<IResultObject>

Specifies a boot image object. To obtain a boot image object, use the **Get-CMBootImage** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-BootImageId<String[]>

Specifies an array of IDs of boot images.

Aliases	none
---------	------

Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-BootImageName<String[]>

Specifies an array of names of boot images.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeploymentPackage<IResultObject>

Specifies a deployment package object. To obtain a deployment package object, use the **Get-CMDeploymentPackage** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeploymentPackageld<String[]>

Specifies an array of IDs of deployment packages.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeploymentPackageName<String[]>

Specifies an array of names of deployment packages.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Description<String>

Specifies a description for the content.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DisableExportAllDependencies

Indicates that Configuration Manager disables all dependencies for the exported content.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DistributionPointName<String>

Specifies a distribution point for the content.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DriverPackage<IResultObject>

Specifies a driver package object. To obtain a driver package object, use the **Get-CMDriverPackage** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DriverPackageld<String[]>

Specifies an array of IDs of driver packages.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DriverPackageName<String[]>

Specifies an array of names of driver packages.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-FileName<String>

Specifies a file name for a .pkgx file.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OperatingSystemImage<IResultObject>

Specifies an operating system image object. To obtain an operating system image object, use the **Get-
CMOperatingSystemImage** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OperatingSystemImageId<String[]>

Specifies an array of IDs of operating system images.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OperatingSystemImageName<String[]>

Specifies an array of names of operating system images.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OperatingSystemInstaller<IResultObject>

Specifies an operating system installer object. To obtain an operating system installer object, use the **Get-CMOperatingSystemInstaller** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OperatingSystemInstallerId<String[]>

Specifies an array of IDs of operating system installers.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OperatingSystemInstallerName<String[]>

Specifies an array of names of operating system installers.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Package<IResultObject>

Specifies a package object. To obtain a package object, use the **Get-CMPackage** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Packageld<String[]>

Specifies an array of IDs of packages.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PackageName<String[]>

Specifies an array of names of packages.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Publish a package

This command publishes the package that has the ID CM200001 to the specified distribution point as the specified .pkgx file.

```
PS C:\> Publish-CMPrestageContent -PackageId "CM200001" -DistributionPointName  
"FileDist02.Western.Contoso.com" -FileName "C:\Users\admin\Documents\Package.pkgx"
```

Example 2: Publish a boot image

This command publishes the boot image that has the ID CM200005 to the specified distribution point as the specified .pkgx file.

```
PS C:\> Publish-CMPrestageContent -BootImageId "CM200005" -DistributionPointName  
"FileDist02.Western.Contoso.com" -FileName "C:\Users\admin\Documents\BootImage.pkgx"
```

Example 3: Publish a driver package

This command publishes the driver package that has the ID CM20000F to the specified distribution point as the specified .pkgx file.

```
PS C:\> Publish-CMPrestageContent -DriverPackageId "CM20000F" -DistributionPointName  
"FileDist02.Western.Contoso.com" -FileName "C:\Users\admin\Documents\DriverPackage.pkgx"
```

Example 4: Publish an operating system image

This command publishes the operating system image that has the ID CM200006 to the specified distribution point as the specified .pkgx file.

```
PS C:\> Publish-CMPrestageContent -OperatingSystemImageId "CM200006" -DistributionPointName  
"FileDist02.Western.Contoso.com" -FileName "C:\Users\admin\Documents\OSImage.pkgx"
```

Example 5: Publish an operating system installer

This command publishes the operating system installer that has the ID CM200017 to the specified distribution point as the specified .pkgx file.

```
PS C:\> Publish-CMPrestageContent -OperatingSystemInstallerId "CM200017" -  
DistributionPointName "FileDist02.Western.Contoso.com" -FileName  
"C:\Users\admin\Documents\OSInstaller.pkgx"
```

Related topics

[Get-CMApplication](#)

[Get-CMBootImage](#)

[Get-CMDeploymentPackage](#)

[Get-CMDriverPackage](#)

[Get-CMOperatingSystemImage](#)

[Get-CMOperatingSystemInstaller](#)

[Get-CMPackage](#)

Publish-CMPrestageContentTaskSequence

Publish-CMPrestageContentTaskSequence

Distributes the content that a task sequence uses to a distribution point.

Syntax

Parameter Set: SearchByIdMandatory_TaskSequence

```
Publish-CMPrestageContentTaskSequence -DistributionPointName <String> -FolderName <String> -TaskSequenceId <String[]> [-Description <String> ] [-DisableIncludeApplicationDependencies] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByNameMandatory_TaskSequence

```
Publish-CMPrestageContentTaskSequence -DistributionPointName <String> -FolderName <String> -TaskSequenceName <String[]> [-Description <String> ] [-DisableIncludeApplicationDependencies] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory_TaskSequence

```
Publish-CMPrestageContentTaskSequence -DistributionPointName <String> -FolderName <String> -TaskSequence <IResultObject> [-Description <String> ] [-DisableIncludeApplicationDependencies] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Publish-CMPrestageContentTaskSequence** cmdlet distributes the content that a task sequence uses to a distribution point. Optionally, you can exclude the application dependencies for applications indicated in the task sequence.

Parameters

-Description<String>

Specifies a description for the content to prestage.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DisableIncludeApplicationDependencies

Indicates that distributed content excludes application dependencies for applications indicated in the task sequence.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DistributionPointName<String>

Specifies the name of a distribution point that is associated with the task sequence.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-FolderName<String>

Specifies a folder name. The folder that you specify contains prestaged content files.

Aliases	none
Required?	true
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-TaskSequence<IResultObject>

Specifies a task sequence object. To obtain a task sequence object, use the **Get-CMTaskSequence** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-TaskSequenceId<String[]>

Specifies an array of IDs of task sequences.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-TaskSequenceName<String[]>

Specifies an array of names of task sequences.

Aliases	none
Required?	true

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

[Get-CMTaskSequence](#)

[Publish-CMPrestageContent](#)

Remove-CMAccessAccount

Remove-CMAccessAccount

Removes users or groups from an access account.

Syntax

Parameter Set: SearchByApplicationName

```
Remove-CMAccessAccount -AccountType {Administrator | Guest | User | WindowsGroup | WindowsUser} -ApplicationName <String> [-Force] [-UserName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByApplication

```
Remove-CMAccessAccount -AccountType {Administrator | Guest | User | WindowsGroup | WindowsUser} -Application <IResultObject> [-Force] [-UserName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByApplicationId

```
Remove-CMAccessAccount -AccountType {Administrator | Guest | User | WindowsGroup | WindowsUser} -ApplicationId <String> [-Force] [-UserName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByBootImage

```
Remove-CMAccessAccount -AccountType {Administrator | Guest | User | WindowsGroup | WindowsUser} -BootImage <IResultObject> [-Force] [-UserName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByBootImageId

```
Remove-CMAccessAccount -AccountType {Administrator | Guest | User | WindowsGroup | WindowsUser} -BootImageId <String> [-Force] [-UserName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByBootImageName

```
Remove-CMAccessAccount -AccountType {Administrator | Guest | User | WindowsGroup | WindowsUser} -BootImageName <String> [-Force] [-UserName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByDriverPackage

```
Remove-CMAccessAccount -AccountType {Administrator | Guest | User | WindowsGroup | WindowsUser} -DriverPackage <IResultObject> [-Force] [-UserName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByDriverPackageId

```
Remove-CMAccessAccount -AccountType {Administrator | Guest | User | WindowsGroup | WindowsUser} -DriverPackageId <String> [-Force] [-UserName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByDriverPackageName

Remove-CMAccessAccount -AccountType {Administrator | Guest | User | WindowsGroup | WindowsUser} -DriverPackageName <String> [-Force] [-UserName <String>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SearchByOSImage

Remove-CMAccessAccount -AccountType {Administrator | Guest | User | WindowsGroup | WindowsUser} -OperatingSystemImage <IResultObject> [-Force] [-UserName <String>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SearchByOSImageId

Remove-CMAccessAccount -AccountType {Administrator | Guest | User | WindowsGroup | WindowsUser} -OperatingSystemImageId <String> [-Force] [-UserName <String>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SearchByOSImageName

Remove-CMAccessAccount -AccountType {Administrator | Guest | User | WindowsGroup | WindowsUser} -OperatingSystemImageName <String> [-Force] [-UserName <String>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SearchByOSInstaller

Remove-CMAccessAccount -AccountType {Administrator | Guest | User | WindowsGroup | WindowsUser} -OperatingSystemInstaller <IResultObject> [-Force] [-UserName <String>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SearchByOSInstallerId

Remove-CMAccessAccount -AccountType {Administrator | Guest | User | WindowsGroup | WindowsUser} -OperatingSystemInstallerId <String> [-Force] [-UserName <String>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SearchByOSInstallerName

Remove-CMAccessAccount -AccountType {Administrator | Guest | User | WindowsGroup | WindowsUser} -OperatingSystemInstallerName <String> [-Force] [-UserName <String>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SearchByPackage

Remove-CMAccessAccount -AccountType {Administrator | Guest | User | WindowsGroup | WindowsUser} -Package <IResultObject> [-Force] [-UserName <String>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SearchByPackageId

Remove-CMAccessAccount -AccountType {Administrator | Guest | User | WindowsGroup | WindowsUser} -PackageId <String> [-Force] [-UserName <String>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SearchByPackageName

Remove-CMAccessAccount -AccountType {Administrator | Guest | User | WindowsGroup | WindowsUser} -PackageName <String> [-Force] [-UserName <String>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SearchBySoftwareUpdateDeploymentPackage

Remove-CMAccessAccount -AccountType {Administrator | Guest | User | WindowsGroup |

```
WindowsUser} -SoftwareUpdateDeploymentPackage <IResultObject> [-Force] [-UserName <String> ]
[-Confirm] [-WhatIf] [ <CommonParameters>]
```

```
Parameter Set: SearchBySoftwareUpdateDeploymentPackageId
Remove-CMAccessAccount -AccountType {Administrator | Guest | User | WindowsGroup |
WindowsUser} -SoftwareUpdateDeploymentPackageId <String> [-Force] [-UserName <String> ] [-
Confirm] [-WhatIf] [ <CommonParameters>]
```

```
Parameter Set: SearchBySoftwareUpdateDeploymentPackageName
Remove-CMAccessAccount -AccountType {Administrator | Guest | User | WindowsGroup |
WindowsUser} -SoftwareUpdateDeploymentPackageName <String> [-Force] [-UserName <String> ] [-
Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Remove-CMAccessAccount** cmdlet removes users or groups from an access account.

An access account is a list of users or groups that can access an established service or application that is located on a distribution point. For example, members in the Software Update Point Connection Access Account can access two services to manage software updates: Windows Server Update Services (WSUS) and WSUS Synchronization Manager.

Parameters

-AccountType<AccessAccountType>

Specifies an account type. Valid values are: Guest, User, WindowsGroup, and WindowsUser.

The acceptable values for this parameter are:

Administrator	
Guest	
User	
WindowsGroup	
WindowsUser	

Aliases	none
Required?	true
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Application<IResultObject>

Specifies a deployed application object. You can get an application object by using the **Get-
CMAApplication** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ApplicationId<String>

Specifies the ID of an application.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ApplicationName<String>

Specifies the name of an application.

Aliases	none
Required?	true

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-BootImage<IResultObject>

Specifies a boot image object. A boot image object contains the Windows files that are required to prepare a computer for the installation of an operating system. You can get a boot image object by using the **Get-CMBootImage** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-BootImageId<String>

Specifies the ID of a boot image.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-BootImageName<String>

Specifies the name of a boot image.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DriverPackage<IResultObject>

Specifies a driver package object. A driver package object specifies a group of hardware drivers that are required to install an operating system. You can get a driver package object by using the **Get-CMDriverPackage** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DriverPackageld<String>

Specifies the ID of a driver package.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DriverPackageName<String>

Specifies the name of a driver package.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OperatingSystemImage<IResultObject>

Specifies an operating system image object. An operating system image object contains the Windows files that compose a complete Windows installation. You can get an operating system image object by using the **Get-CMOperatingSystemImage** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OperatingSystemImageId<String>

Specifies the ID of an operating system image.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OperatingSystemImageName<String>

Specifies the name of an operating system image.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OperatingSystemInstaller<IResultObject>

Specifies an operating system installer object. An operating system installer object contains the Windows files that are required to prepare a computer for the installation of an operating system. To obtain an operating system installer object, use the **Get-CMOperatingSystemInstaller** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OperatingSystemInstallerId<String>

Specifies the ID of an operating system installer.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OperatingSystemInstallerName<String>

Specifies the name of an operating system installer.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Package<IResultObject>

Specifies a deployed software script or program object. You can get a package by using the **Get-CMPackage** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-PackageId<String>

Specifies the ID of a deployed software script or program.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PackageName<String>

Specifies the name of a deployed software script or program.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SoftwareUpdateDeploymentPackage<IResultObject>

Specifies a software update deployment object. You can get a software update deployment object by using the **Get-CMSoftwareUpdateDeploymentPackage** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SoftwareUpdateDeploymentPackageId<String>

Specifies the ID of a deployed software update.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SoftwareUpdateDeploymentPackageName<String>

Specifies the name of a deployed software update.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UserName<String>

Specifies a Windows user account name in *domain\user* format.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Remove a user from an access account for an application that is specified by using its name

This command removes a Windows user from the access account for an application that is specified by using its name. You must confirm the action before the command performs it.

```
PS C:\> Remove-CMAccessAccount -ApplicationName "SharePoint 2010" -Type WindowsUser -
UserName "CONTOSO\ENarvaez" -Confirm
```

Example 2: Remove a group from an access account for a package that is specified by using its ID

In this example, the first command gets the package object ID and stores it in the variable \$ID. The second command removes a group from the access account for the identified package. No confirmation is required.

```
PS C:\> $ID = Get-CMAccessAccount -PackageId "CM1100002"
PS C:\> Remove-CMAccessAccount - PackageId $ID -Type WindowsGroup -UserName "CONTOSO\Guest"
```

Related topics

[New-CMAccessAccount](#)

[Get-CMAccessAccount](#)

[Set-CMAccessAccount](#)

[Get-CMApplication](#)

[Get-CMBootImage](#)

[Get-CMDriverPackage](#)

[Get-CMOperatingSystemImage](#)

[Get-CMOperatingSystemInstaller](#)

[Get-CMPackage](#)

[Get-CMSoftwareUpdateDeploymentPackage](#)

Remove-CMAccount

Remove-CMAccount

Removes a specified user.

Syntax

Parameter Set: SearchByNameMandatory

```
Remove-CMAccount -Name <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Remove-CMAccount -InputObject <IResultObject> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Remove-CMAccount** cmdlet removes a user account from Microsoft System Center 2012 Configuration Manager. System Center 2012 Configuration Manager uses user accounts to connect to various system and network resources. For more information about user accounts, see [Technical Reference for Accounts Used in Configuration Manager](http://go.microsoft.com/fwlink/?LinkID=248317) (<http://go.microsoft.com/fwlink/?LinkID=248317>) on TechNet.

Parameters

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a user account object. You can get a user account object by using the **Get-CMAccount** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies the name of the user account.

Aliases	UserName
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Remove a user account by using its name

This command removes the user account that is specified by its name.

```
PS C:\> Remove-CMAccount -Name "CONTOSO\EDaugherty"
```

Related topics

[Get-CMAccount](#)

[New-CMAccount](#)

[Set-CMAccount](#)

Remove-CMActiveDirectoryForest

Remove-CMActiveDirectoryForest

Removes an Active Directory forest object from Configuration Manager.

Syntax

Parameter Set: SearchByIdMandatory

```
Remove-CMActiveDirectoryForest -Id <String[]> [-Force] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByFQDNMandatory

```
Remove-CMActiveDirectoryForest -ForestFqdn <String> [-Force] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Remove-CMActiveDirectoryForest -InputObject <IResultObject> [-Force] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Detailed Description

The **Remove-CMActiveDirectoryForest** cmdlet removes an Active Directory forest object from Microsoft System Center 2012 Configuration Manager. You can specify an Active Directory forest by using the ID property or the fully qualified domain name (FQDN), or you can supply the Active Directory forest itself.

Parameters

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-ForestFqdn<String>

Specifies the FQDN of a Configuration Manager object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Id<String[]>

Specifies an array of IDs of Configuration Manager objects. You can find the identifier value in the ForestID property of an Active Directory forest.

Aliases	ForestId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies an Active Directory forest object in Configuration Manager.

Aliases	none
Required?	true
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Remove an Active Directory forest object by ID

This command removes an Active Directory forest object that has the ID 16777217.

```
PS C:\> Remove-CMActiveDirectoryForest -Id "16777217"
```

Related topics

[New-CMActiveDirectoryForest](#)

[Get-CMActiveDirectoryForest](#)

[Set-CMActiveDirectoryForest](#)

Remove-CMAdministrativeUser

Remove-CMAdministrativeUser

Removes Configuration Manager administrative users.

Syntax

Parameter Set: SearchByIdMandatory

```
Remove-CMAdministrativeUser -Id <String[]> [-Force] [-RoleName <String[]> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByNameMandatory

```
Remove-CMAdministrativeUser -Name <String[]> [-Force] [-RoleName <String[]> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Remove-CMAdministrativeUser -InputObject <IResultObject> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Remove-CMAdministrativeUser** cmdlet removes one or more Microsoft System Center 2012 Configuration Manager administrative users. When you remove an administrative user, System Center 2012 Configuration Manager revokes the access of the administrative user to manage Configuration Manager.

Parameters

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-Id<String[]>

Specifies an array of IDs for administrative users.

Aliases	AdminId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a **CMAdministrativeUser** object. To obtain a **CMAdministrativeUser** object, use the **Get-
CMAdministrativeUser** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String[]>

Specifies an array of administrative user names in the form of *<domain>\<user>*.

Aliases	DisplayName
Required?	true
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	true

-RoleName<String[]>

Specifies an array of names of security roles. Valid values are:

- Application Administrator
- Application Author
- Application Deployment Manager
- Asset Manager
- Compliance Settings Manager
- Discovery Operator
- Endpoint Protection Manager
- Full Administrator
- Infrastructure Administrator
- Operating System Deployment Manager
- Operations Administrator
- Read-only Analyst
- Remote Tools Operator
- Security Administrator
- Software Update Manager
- custom-defined security roles

Aliases	RoleNames
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Remove an administrative user by using a wildcard

This command removes administrative users who have the first name Evan.

```
PS C:\> Remove-CMAdministrativeUser -Name Evan*
```

Example 2: Remove an administrative user by using an ID

This command removes the administrative user who has the identity 16777225.

```
PS C:\> Remove-CMAdministrativeUser -Id "16777225"
```

Example 3: Remove an administrative user by using an object variable

In this example, the first command gets administrative users who have the name Evan and stores them in the \$AdminUser variable.

The second command removes the administrative users stored in the variable \$AdminUser.

```
PS C:\> $AdminUser = Get-CMAdministrativeUser -Name Evan*
PS C:\> Remove-CMAdministrativeUser -InputObject $AdminUser
```

Related topics

[Get-CMAdministrativeUser](#)

[New-CMAdministrativeUser](#)

Remove-CMAAlert

Remove-CMAAlert

Removes Configuration Manager alerts.

Syntax

Parameter Set: SearchByIdMandatory

```
Remove-CMAAlert -Id <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByNameMandatory

```
Remove-CMAAlert -Name <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Remove-CMAAlert -InputObject <IResultObject> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Remove-CMAAlert** cmdlet removes one or more Microsoft System Center 2012 Configuration Manager alerts.

Parameters

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Id<String>

Specifies an alert identifier. You can obtain the identifier of an alert by using the **Get-Alert** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a **CMAAlert** object. To obtain a **CMAAlert** object, use the **Get-CMAAlert** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies an alert name. You can obtain the name of an alert by using **Get-CMAAlert**.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Remove an alert by using alert ID

This command removes an alert that has the ID16777223.

```
PS C:\> Remove-CMAAlert -Id "16777223"
```

Example 2: Remove an alert by using alert object variable

In this example, the first command gets a **CMAAlert** object that has the ID 16777221 and stores it in the \$AlertObj variable.

The second command removes the alert stored in the \$AlertObj variable.

```
PS C:\> $AlertObj = Get-CMAAlert -Id "16777221" PS C:\> Remove-  
CMAAlert -InputObject $AlertObj
```

Related topics

[Enable-CMAAlert](#)

[Get-CMAAlert](#)

[Set-CMAAlert](#)

[Suspend-CMAAlert](#)

[Disable-CMAAlert](#)

Remove-CMAAlertSubscription

Remove-CMAAlertSubscription

Removes an alert subscription object.

Syntax

Parameter Set: SearchByIdMandatory

```
Remove-CMAAlertSubscription -Id <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByNameMandatory

```
Remove-CMAAlertSubscription -Name <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Remove-CMAAlertSubscription -InputObject <IResultObject> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Remove-CMAAlertSubscription** cmdlet removes an alert subscription from Microsoft System Center 2012 Configuration Manager.

Parameters

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Id<String>

Specifies an array of identifiers of subscriptions.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies an alert notification object in Configuration Manager.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies the name of an alert subscription object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Remove an alert subscription by ID

This command removes an alert subscription by using its ID.

```
PS C:\> Remove-CMAAlertSubscription -Id "16777310"
```

Example 2: Remove an alert subscription by name

This command removes an alert subscription named Subscription01.

```
PS C:\> Remove-CMAAlertSubscription -Name "Subscription01"
```

Example 3: Remove an alert subscription by using the output from another cmdlet as input

The first command gets an alert subscription object that has the ID 16777310 and stores the object in the \$SubObj variable.

The second command deletes the alert subscription that is stored in the \$SubObj variable.

```
PS C:\> $SubObj = Get-CMAAlertSubscription -Id "16777310"
```

```
PS C:\> Remove-CMAAlertSubscription -AlertSubscription $SubObj
```

Related topics

[New-CMAAlertSubscription](#)

[Get-CMAAlertSubscription](#)

[Set-CMAAlertSubscription](#)

Remove-CMAmtProvisioningData

Remove-CMAmtProvisioningData

Removes provisioning information for an Intel AMT computer.

Syntax

Parameter Set: SearchByNameMandatory

```
Remove-CMAmtProvisioningData -ControlType {FullUnprovision | FullUnprovisionSuppressAuto | KerberosFullUnprovision | KerberosFullUnprovisionSuppressAuto | KerberosPartialUnprovision | KerberosPartialUnprovisionSuppressAuto | PartialUnprovision | PartialUnprovisionSuppressAuto} -DeviceName <String[]> [-Force] [-Confirm] [-WhatIf] [<CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Remove-CMAmtProvisioningData -ControlType {FullUnprovision | FullUnprovisionSuppressAuto | KerberosFullUnprovision | KerberosFullUnprovisionSuppressAuto | KerberosPartialUnprovision | KerberosPartialUnprovisionSuppressAuto | PartialUnprovision | PartialUnprovisionSuppressAuto} -DeviceId <String[]> [-Force] [-Confirm] [-WhatIf] [<CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Remove-CMAmtProvisioningData -ControlType {FullUnprovision | FullUnprovisionSuppressAuto | KerberosFullUnprovision | KerberosFullUnprovisionSuppressAuto | KerberosPartialUnprovision | KerberosPartialUnprovisionSuppressAuto | PartialUnprovision | PartialUnprovisionSuppressAuto} -Device <IResultObject> [-Force] [-Confirm] [-WhatIf] [<CommonParameters>]
```

Detailed Description

The **Remove-CMAmtProvisioningData** cmdlet removes the Microsoft System Center 2012 Configuration Manager provisioning information for an Intel Active Management Technology (Intel AMT)-based computer. You might want to remove provisioning information when you no longer want to manage a computer out of band by using System Center 2012 Configuration Manager.

You can either remove all configuration data or retain identification information about the computer, such as its host name, IP address, and DNS suffix.

By default, System Center 2012 Configuration Manager automatically reprovisions Intel AMT-based computers if you have configured Intel AMT provisioning. Use the *ControlType* parameter to control reprovisioning for individual computers.

Parameters

-ControlType<RemoteControlType>

Specifies how Configuration Manager removes provisioning information. This parameter controls whether the cmdlet removes some or all provisioning information. This parameter also determines whether the cmdlet allows Configuration Manager to reprovision an Intel AMT-based computer later.

Valid values are:

- FullUnprovision. Resets Intel AMT to the factory default settings.
- FullUnprovisionSuppressAuto. Resets Intel AMT to the factory default settings and does not allow Configuration Manager to reprovision the computer.
- KerberosFullUnprovision. Resets Intel AMT for Kerberos-enabled computers to factory default settings.
- KerberosFullUnprovisionSuppressAuto. Resets Intel AMT for Kerberos-enabled computers to factory default settings, and does not allow Configuration Manager to reprovision the computer.
- KerberosPartialUnprovision. Resets Intel AMT for Kerberos-enabled computers except for identification information about the computer.
- KerberosPartialUnprovisionSuppressAuto. Resets Intel AMT for Kerberos-enabled computers except for identification information about the computer, and does not allow Configuration Manager to reprovision the computer.
- PartialUnprovision. Resets Intel AMT to the factory default settings, except for identification information about the computer.
- PartialUnprovisionSuppressAuto. Resets Intel AMT to the factory default settings, except for identification information about the computer, and does not allow Configuration Manager to reprovision the computer.

The acceptable values for this parameter are:

FullUnprovision	
FullUnprovisionSuppressAuto	
KerberosFullUnprovision	
KerberosFullUnprovisionSuppressAuto	
KerberosPartialUnprovision	
KerberosPartialUnprovisionSuppressAuto	
PartialUnprovision	
PartialUnprovisionSuppressAuto	

Aliases	none
---------	------

Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Device<IResultObject>

Specifies a device object. To obtain a device object, use the **Get-CMDevice** cmdlet.

Aliases	InputObject
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeviceId<String[]>

Specifies an array of IDs of devices.

Aliases	ResourceId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeviceName<String[]>

Specifies an array of names of devices.

Aliases	Name
---------	------

Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Remove provisioning information completely for a specified computer

This command removes provisioning information from a computer that has the ID SMS000076. The cmdlet removes all provisioning data and suppresses automatic reprovisioning. The command uses the *Force* parameter. Therefore, it does not prompt you for confirmation.

```
PS C:\> Remove-CMAmtProvisioningData -ControlType FullUnprovisionSuppressAuto -DeviceId "SMS000076" -Force
```

Related topics

[Invoke-CMAmtProvisioningDiscovery](#)

[Update-CMAMTProvisioning](#)

[Get-CMDevice](#)

Remove-CMAntiMalwarePolicy

Remove-CMAntiMalwarePolicy

Removes an antimalware policy for Endpoint Protection.

Syntax

Parameter Set: SearchByIdMandatory

```
Remove-CMAntiMalwarePolicy -Id <String> [-Force] [-SecuredScopeNames <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByNameMandatory

```
Remove-CMAntiMalwarePolicy -Name <String> [-Force] [-SecuredScopeNames <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Remove-CMAntiMalwarePolicy -InputObject <IResultObject> [-Force] [-SecuredScopeNames <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Remove-CMAntiMalwarePolicy** cmdlet removes an antimalware policy for System Center 2012 Endpoint Protection from Microsoft System Center 2012 Configuration Manager.

When you create a new antimalware policy for Endpoint Protection and deploy it to a collection of client computers, this antimalware policy overrides the default antimalware policy. When you remove an antimalware policy for Endpoint Protection, System Center 2012 Configuration Manager removes the antimalware agent from the client computers and applies the policy with the next highest priority. If no other policies exist, then the default antimalware policy will be applied.

Parameters

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Id<String>

Specifies the ID of an antimalware policy object.

Aliases	SettingsId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a **CMAntiMalwarePolicy** object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies the name of an antimalware policy.

Aliases	none
Required?	true
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SecuredScopeNames<String>

Specifies the names of secured scopes.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Remove an antimalware policy by using ID

This command removes the antimalware policy that has the ID 16777217.

```
PS C:\> Remove-CMAntiMalwarePolicy -Id "16777217"
```

Example 2: Remove an antimalware policy by using a wildcard

This command removes antimalware policies that have a name that begins with the letter D.

```
PS C:\> Remove-CMAntiMalwarePolicy -Name D*
```

Example 3: Remove an antimalware policy by using object variable

In this example, the first command gets an antimalware policy object that has the ID 16777217 and stores it in the \$AMPobj variable.

The second command removes the antimalware policy stored in the \$AMPobj variable.

```
PS C:\> $AMPobj = Get-CMAntiMalwarePolicy -Id "16777217"
```

```
PS C:\> Remove-CMAntiMalwarePolicy -InputObject $AMPobj
```

Related topics

[Export-CMAntimalwarePolicy](#)

[Get-CMAntiMalwarePolicy](#)

[Import-CMAntimalwarePolicy](#)

[Merge-CMAntimalwarePolicy](#)

[New-CMAntimalwarePolicy](#)

[Set-CMAntiMalwarePolicy](#)

Remove-CMApplication

Remove-CMApplication

Removes an application in Configuration Manager.

Syntax

Parameter Set: SearchByNameMandatory

```
Remove-CMApplication -Name <String[]> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Remove-CMApplication -Id <String[]> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Remove-CMApplication -InputObject <IResultObject> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Remove-CMApplication** cmdlet removes an application so that it cannot be installed by clients. This cmdlet does not remove any existing client installations.

Parameters

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Id<String[]>

Specifies an array of IDs for the applications.

Aliases	CId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies an application object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String[]>

Specifies an array of names for the application.

Aliases	LocalizedDisplayName
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Remove an application by using name

This command removes an application by name.

```
PS C:\> Remove-CMApplication -Name "Contoso-test"
```

Related topics

[Export-CMApplication](#)

[Get-CMApplication](#)

[Import-CMApplication](#)

[New-CMApplication](#)

[Resume-CMApplication](#)

[Set-CMApplication](#)

[Suspend-CMApplication](#)

Remove- CMApplicationCatalogWebServicePoint

Remove-CMApplicationCatalogWebServicePoint

Removes an Application Catalog web service point.

Syntax

Parameter Set: SearchByNameMandatory

```
Remove-CMApplicationCatalogWebServicePoint -SiteCode <String> -SiteSystemServerName <String>  
[-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Remove-CMApplicationCatalogWebServicePoint -InputObject <IResultObject> [-Force] [-Confirm]  
[-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Remove-CMApplicationCatalogWebServicePoint** cmdlet removes a Microsoft System Center 2012 Configuration Manager Application Catalog web service point object that has a specified site code for a fully qualified domain name (FQDN).

Parameters

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies an Application Catalog web service point object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteCode<String>

Specifies a Configuration Manager site code for an Application Catalog web service point.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteSystemServerName<String>

Specifies an FQDN for an application catalog web service point.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Remove a system role

This command removes an Application Catalog web service point named western.contoso.com that has the site code CM1.

```
PS C:\> Remove-CMApplicationCatalogWebServicePoint -SiteSystemServerName  
"western.contoso.com" -SiteCode "CM1"
```

Related topics

[Add-CMApplicationCatalogWebServicePoint](#)

[Get-CMApplicationCatalogWebServicePoint](#)

Remove-CMApplicationCatalogWebSitePoint

Remove-CMApplicationCatalogWebSitePoint

Removes a Configuration Manager Application Catalog website point.

Syntax

Parameter Set: SearchByNameMandatory

```
Remove-CMApplicationCatalogWebSitePoint -SiteCode <String> -SiteSystemServerName <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Remove-CMApplicationCatalogWebSitePoint -InputObject <IResultObject> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Remove-CMApplicationCatalogWebSitePoint** cmdlet removes an Application Catalog website point in Microsoft System Center 2012 Configuration Manager. This site system role supports the Application Catalog website and the Software Library.

You can specify a website point to remove by site code and name of the server that hosts the role, or you can use the **Get-CMApplicationCatalogWebsitePoint** cmdlet to get a website point to remove.

Parameters

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies an Application Catalog website point object. To obtain Application Catalog website point object, use the **Get-CMApplicationCatalogWebsitePoint** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteCode<String>

Specifies the site code for a Configuration Manager site.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteSystemServerName<String>

Specifies the name of a server that hosts a site system role.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Remove an Application Catalog website point

This command removes an Application Catalog website point that belongs to the site that has the site code CM2. The computer named WesternACWP.Contoso.com hosts the point that the cmdlet removes.

```
PS C:\> Remove-CAApplicationCatalogWebsitePoint -SiteCode "CM2" -SiteSystemServerName "WesternACWP.Contoso.com"
```

Example 2: Remove an Application Catalog website point by using a variable

The first command uses the **Get-CMApplicationCatalogWebsitePoint** cmdlet to get an Application Catalog website point hosted by the computer named WesternACWP.Contoso.com in the site that has the site code CM2, and stores it in the \$CMACWP variable.

The second command removes the Application Catalog website point stores in the \$CMACWP variable. The command includes the *Force* parameter. Therefore, the command does not prompt you for confirmation.

```
PS C:\> $CMACWP= Get-CMApplicationCatalogWebsitePoint -SiteCode "CM2" -  
SiteSystemServerName"WesternACWP.Contoso.com"  
PS C:\> Remove-CMApplicationCatalogWebsitePoint -InputObject $CMACWP -Force
```

Related topics

[Add-CMApplicationCatalogWebsitePoint](#)

[Get-CMApplicationCatalogWebsitePoint](#)

[Set-CMApplicationCatalogWebsitePoint](#)

Remove-CMApplicationRevisionHistory

Remove-CMApplicationRevisionHistory

Removes a revision history from a Configuration Manager application.

Syntax

Parameter Set: SearchByNameMandatory

```
Remove-CMApplicationRevisionHistory -Name <String[]> -Revision <Int32> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Remove-CMApplicationRevisionHistory -Id <String[]> -Revision <Int32> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Remove-CMApplicationRevisionHistory -InputObject <IResultObject> -Revision <Int32> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Remove-CMApplicationRevisionHistory** cmdlet removes a revision history from a Microsoft System Center 2012 Configuration Manager application. The revision history contains a list of revisions to an application or a development type that the application contains.

Parameters

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Id<String[]>

Specifies an array of IDs that identify the application revision histories that you delete.

Aliases	Ciid
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies an application object. To obtain an application object, use the **Get-CMApplication** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String[]>

Specifies an array of names for the application revision histories that you delete.

Aliases	LocalizedDisplayName
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Revision<Int32>

Specifies the version number of the revision that you delete from the history.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Remove a revision history

This command removes the revision history named MSXML 6.0 Parser.

```
PS C:\> Remove-CMApplicationRevisionHistory -Name "MSXML 6.0 Parser"
```

Related topics

[Get-CMApplicationRevisionHistory](#)

[Restore-CMApplicationRevisionHistory](#)

Remove-CMAppVVirtualEnvironment

Remove-CMAppVVirtualEnvironment

Removes an App-V virtual environment.

Syntax

Parameter Set: SearchByIdMandatory

```
Remove-CMAppVVirtualEnvironment -Id <String[]> [-Force] [-SecuredScopeNames <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByNameMandatory

```
Remove-CMAppVVirtualEnvironment -Name <String[]> [-Force] [-SecuredScopeNames <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Remove-CMAppVVirtualEnvironment -InputObject <IResultObject> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Remove-CMAppVVirtualEnvironment** cmdlet removes one or more Microsoft Application Virtualization (App-V) virtual environment objects from Microsoft System Center 2012 Configuration Manager. You can specify App-V virtual environments by name or ID, or you can provide an App-V virtual environment object.

Parameters

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-Id<String[]>

Specifies an array of IDs of virtual environments.

Aliases	Ciid
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies an App-V virtual environment object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String[]>

Specifies an array of names of App-V virtual environment objects. You can use a wildcard.

Aliases	LocalizedDisplayName
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-SecuredScopeNames<String>

Specifies the name of security scopes. A security scope can be Default or the name of a custom-created security scope.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Remove a virtual environment by name

This command removes an App-V virtual environment named Test.

```
PS C:\> Remove-CMAppVVirtualEnvironment -Name "Test"
```

Example 2: Remove a virtual environment by ID

This command removes an App-V virtual environment that has the ID 16781806.

```
PS C:\> Remove-CMAppVVirtualEnvironment -Id "16781806"
```

Example 3: Remove a virtual environment by name by using a wildcard

The first command gets all App-V virtual environments that have names that begin with the letter T and stores them in the \$AppV variable.

The second command removes all the environments stored in \$AppV.

```
PS C:\> $AppV = Get-CMAppVVirtualEnvironment -Name "T*"
```

```
PS C:\> Remove-CMAppVVirtualEnvironment -InputObject $AppV
```

Related topics

[Get-CMAppVVirtualEnvironment](#)

[New-CMAppVVirtualEnvironment](#)

[Set-CMAppVVirtualEnvironment](#)

Remove-CMAAssetIntelligenceCatalogItem

Remove-CMAAssetIntelligenceCatalogItem

Removes an item from the Asset Intelligence catalog.

Syntax

Parameter Set: SearchByIdMandatory

```
Remove-CMAAssetIntelligenceCatalogItem -Id <String[]> [-Force] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByNameMandatory

```
Remove-CMAAssetIntelligenceCatalogItem -CategoryName <String> [-Force] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Remove-CMAAssetIntelligenceCatalogItem -InputObject <IResultObject> [-Force] [-Confirm] [-  
WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Remove-CMAAssetIntelligenceCatalogItem** cmdlet removes software categories, software families, and custom software labels from the Asset Intelligence catalog in Microsoft System Center 2012 Configuration Manager.

The Asset Intelligence catalog contains categorization and identification information for software titles. The catalog includes predefined categories and families. Predefined items cannot be modified. In addition to predefined software categories and software families, you can create custom categories and families. You can also create custom software labels.

Parameters

-CategoryName<String>

Specifies the name of a category, family, or label in the Asset Intelligence catalog.

Aliases	none
Required?	true
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Id<String[]>

Specifies an array of IDs of asset intelligence catalog items.

Aliases	CategoryId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies an Asset Intelligence catalog item. To obtain an Asset Intelligence catalog item, use the **Get-CMAssetIntelligenceCatalogItem** cmdlet.

Aliases	none
Required?	true

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Remove a catalog item by category name

This command removes the category named Database Tools from the Asset Intelligence catalog.

PS C:\> Remove-CMAssetIntelligenceCatalogItem -CategoryName "Database Tools"

Related topics

[New-CMAssetIntelligenceCatalogItem](#)

[Get-CMAssetIntelligenceCatalogItem](#)

[Set-CMAssetIntelligenceCatalogItem](#)

Remove-CMAssetIntelligenceSynchronizationPoint

Remove-CMAssetIntelligenceSynchronizationPoint

Removes an Asset Intelligence synchronization point.

Syntax

Parameter Set: SearchByNameMandatory

```
Remove-CMAssetIntelligenceSynchronizationPoint -SiteCode <String> -SiteSystemServerName <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Remove-CMAssetIntelligenceSynchronizationPoint -InputObject <IResultObject> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Remove-CMAssetIntelligenceSynchronizationPoint** cmdlet removes an Asset Intelligence synchronization point from a site system. After you remove an Asset Intelligence synchronization point, the Microsoft System Center 2012 Configuration Manager sites that used the synchronization point cannot connect to System Center Online.

Parameters

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies an Asset Intelligence synchronization point object. To obtain a **CMAssetIntelligenceSynchronizationPoint** object, use the **Get-
CMAssetIntelligenceSynchronizationPoint** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteCode<String>

Specifies the three-letter site code of the Configuration Manager site that hosts the site system role.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteSystemServerName<String>

Specifies an array of fully qualified domain names (FQDN) of the servers that host the site system role.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

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

Example 1: Remove an Asset Intelligence synchronization point

This command removes the Asset Intelligence synchronization point on the System Center 2012 Configuration Manager site that has the site code CM1 on the site system server named CMDIV-WEST04.CORP.CONTOSO.COM.

```
PS C:\> Remove-CMAssetIntelligenceSynchronizationPoint -SiteSystemServerName "CMDIV-  
WEST04.CORP.CONTOSO.COM" -SiteCode "CM1"
```

Example 2: Remove an Asset Intelligence synchronization point by using an object variable

The first command gets the Asset Intelligence synchronization point on the System Center 2012 Configuration Manager site that has the site code ST1 on the site system server named CMDIV-WEST04.CORP.CONTOSO.COM. The command stores the results in the \$AIsync variable.

The second command removes the Asset Intelligence synchronization point stored in the \$AIsync variable.

```
PS C:\> $AIsync = Get-CMAAssetIntelligenceSynchronizationPoint -SiteSystemServerName  
"WEST04.CORP.CONTOSO.COM" -SiteCode "ST1"  
PS C:\> Remove-CMAAssetIntelligenceSynchronizationPoint -InputObject $AIsync
```

Related topics

[Add-CMAAssetIntelligenceSynchronizationPoint](#)

[Get-CMAAssetIntelligenceSynchronizationPoint](#)

[Set-CMAAssetIntelligenceSynchronizationPoint](#)

Remove-CMBaseline

Remove-CMBaseline

Removes configuration baselines.

Syntax

Parameter Set: SearchByIdMandatory

```
Remove-CMBaseline -Id <String[]> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByNameMandatory

```
Remove-CMBaseline -Name <String[]> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Remove-CMBaseline -InputObject <IResultObject> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Remove-CMBaseline** cmdlet removes one or more configuration baseline items in Microsoft System Center 2012 Configuration Manager. You must remove all references to a configuration baseline before you can remove the configuration baseline. After you remove a configuration baseline, System Center 2012 Configuration Manager removes the configuration baseline from the collection of devices to which you deployed it, and Configuration Manager no longer assesses their compliance with the configuration baseline.

Parameters

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-Id<String[]>

Specifies an array of IDs of configuration baselines.

Aliases	Ciid
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a **CMBaseline** object. To obtain a **CMBaseline** object, use the **Get-CMBaseline** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String[]>

Specifies an array of names of configuration baselines.

Aliases	LocalizedDisplayName
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

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

Example 1: Remove a baseline configuration by using a name

This command removes the configuration baseline named BLConfigContoso02.

```
PS C:\> Remove-CMBaseline -Name "BLConfigContoso02"
```

Example 2: Remove a baseline configuration by using an ID

This command removes the configuration baseline that has the ID 16777366.

```
PS C:\> Remove-CMBaseline -Id "16777366"
```

Related topics

[Disable-CMBaseline](#)

[Enable-CMBaseline](#)

[Export-CMBaseline](#)

[Get-CMBaseline](#)

[Import-CMBaseline](#)

[New-CMBaseline](#)

[Set-CMBaseline](#)

[Get-CMBaselineXMLDefinition](#)

[Get-CMBaselineSummarizationSchedule](#)

Remove-CMBootImage

Remove-CMBootImage

Removes an operating system boot image.

Syntax

Parameter Set: SearchByIdMandatory

```
Remove-CMBootImage -Id <String[]> [-Force] [-SecuredScopeNames <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByNameMandatory

```
Remove-CMBootImage -Name <String> [-Force] [-SecuredScopeNames <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Remove-CMBootImage -InputObject <IResultObject> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Remove-CMBootImage** cmdlet removes a Windows Preinstallation Environment (Windows PE) operating system boot image from Microsoft System Center 2012 Configuration Manager.

You must run **Remove-CMBootImage** on the computer that is running the Systems Management Server (SMS) provider. The computer account of the computer that is running the SMS provider must have Read and Write access to the package source of the boot image. For more information about the SMS provider, see [Planning for the SMS Provider in Configuration Manager](http://go.microsoft.com/fwlink/?LinkID=263566) (<http://go.microsoft.com/fwlink/?LinkID=263566>) on TechNet.

Parameters

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Id<String[]>

Specifies an array of boot images identifiers. You can get a boot image object by using the **Get-CMBootImage** cmdlet.

Aliases	PackageId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a boot image object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies a name of a boot image.

Aliases	none
Required?	true

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SecuredScopeNames<String>

Specifies a secured scope name.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Remove a boot image object that is identified by using its ID

This command removes a boot image object that is identified by using its ID. You must confirm the action before the command performs it.

```
PS C:\> Remove-CMBootImage -Id "CM100004" -Confirm
```

Example 2: Remove a boot image object that is identified by using its name

This command removes a boot image object that is identified by using its name. You must confirm the action before the command performs it.

```
PS C:\> Remove-CMBootImage -Name "Boot image (86)" -Confirm
```

Related topics

[Get-CMBootImage](#)

[New-CMBootImage](#)

[Set-CMBootImage](#)

Remove-CMBoundary

Remove-CMBoundary

Removes a boundary.

Syntax

Parameter Set: SearchByIdMandatory

```
Remove-CMBoundary -Id <String[]> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByNameMandatory

```
Remove-CMBoundary -Name <String[]> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Remove-CMBoundary -InputObject <IResultObject> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Remove-CMBoundary** cmdlet removes a boundary from Microsoft System Center 2012 Configuration Manager.

In System Center 2012 Configuration Manager, a boundary is an intranet location that contains one or more devices that you can manage. A boundary can be an IP subnet, Active Directory site name, IPv6 prefix, or an IP address range.

Parameters

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-Id<String[]>

Specifies an array of boundary identifiers (IDs).

Aliases	BoundaryId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies an input object to this cmdlet. You can get the input object by using the **Get-CMBoundary** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String[]>

Specifies an array of boundary names.

Aliases	DisplayName
Required?	true
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Remove a boundary that is specified by its ID

This command removes the boundary that has an identifier of 16777223. Because the *Force* parameter is not specified, you must confirm the action before it is performed.

```
PS C:\> Remove-CMBoundary -Id "16777223"
```

Example 2: Remove a boundary by using an InputObject

In this example, the first command uses the **Get-CMBoundary** cmdlet to get a boundary that has the ID of 16777223, and inserts it into the input object \$BoundaryObj.

The second command identifies the boundary by using the input object \$BoundaryObj and then removes the boundary. Because the *Force* parameter is not specified, you must confirm the action before it is performed.

```
PS C:\> $BoundaryObj = Get-CMBoundary -Id "16777223"  
PS C:\> Remove-Boundary -InputObject $BoundaryObj
```

Related topics

[Get-CMBoundary](#)

[New-CMBoundary](#)

[Set-CMBoundary](#)

[Remove-CMBoundaryFromGroup](#)

[Remove-CMBoundaryGroup](#)

Remove-CMBoundaryFromGroup

Remove-CMBoundaryFromGroup

Removes a Configuration Manager boundary from a boundary group.

Syntax

Parameter Set: RemoveBoundaryFromGroupById_Id

```
Remove-CMBoundaryFromGroup -BoundaryGroupId <Int32> -BoundaryId <Int32> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: RemoveBoundaryFromGroupById_Name

```
Remove-CMBoundaryFromGroup -BoundaryGroupName <String> -BoundaryId <Int32> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: RemoveBoundaryFromGroupById_Object

```
Remove-CMBoundaryFromGroup -BoundaryGroup <IResultObject> -BoundaryId <Int32> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: RemoveBoundaryFromGroupName_Id

```
Remove-CMBoundaryFromGroup -BoundaryGroupId <Int32> -BoundaryName <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: RemoveBoundaryFromGroupName_Name

```
Remove-CMBoundaryFromGroup -BoundaryGroupName <String> -BoundaryName <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: RemoveBoundaryFromGroupName_Object

```
Remove-CMBoundaryFromGroup -BoundaryGroup <IResultObject> -BoundaryName <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: RemoveBoundaryFromGroupByObject_Id

```
Remove-CMBoundaryFromGroup -Boundary <IResultObject> -BoundaryGroupId <Int32> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: RemoveBoundaryFromGroupByObject_Name

```
Remove-CMBoundaryFromGroup -Boundary <IResultObject> -BoundaryGroupName <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: RemoveBoundaryFromGroupByObject_Object

```
Remove-CMBoundaryFromGroup -Boundary <IResultObject> -BoundaryGroup <IResultObject> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Remove-CMBoundaryFromGroup** cmdlet removes a Microsoft System Center 2012 Configuration Manager boundary from a boundary group. A boundary is a network address range, subnet, or Active Directory site that identifies a group of computers that are close in the network. A boundary group is a collection of boundaries.

Parameters

-Boundary<IResultObject>

Specifies a Configuration Manager boundary object to remove. To obtain a boundary object, use the **Get-CMBoundary** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-BoundaryGroup<IResultObject>

Specifies a boundary group object for Configuration Manager. Configuration Manager removes the boundary from this boundary group. To obtain a boundary group object, use the **Get-CMBoundaryGroup** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-BoundaryGroupId<Int32>

Specifies an ID for the boundary group from which you remove a boundary.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-BoundaryGroupName<String>

Specifies a name for the boundary group from which you remove a boundary.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-BoundaryId<Int32>

Specifies an ID for the boundary that you remove.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-BoundaryName<String>

Specifies a name for the boundary that you remove.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Remove a boundary from a group by using the boundary name

This example removes a boundary named CLBound03 from a boundary group that has the ID 16777219.

```
PS C:\> Remove-CMBoundaryFromGroup -BoundaryGroupID "16777219" -BoundaryName "CLBound03"
```

Example 2: Remove multiple boundary groups by using an InputObject

The first command uses the **Get-CMBoundary** cmdlet to get multiple boundaries that are specified by their names, and stores this data into the **\$BoundaryObj** variable.

The second command identifies and removes the boundaries that are specified by using the input object **\$BoundaryObj**. Because the *Force* parameter is not specified, you must confirm the action before it is performed.

```
PS C:\> $BoundaryObj = Get-CMBoundary -Name "Bound01", "Bound02", "Bound03"  
PS C:\> Remove-CMBoundaryFromGroup -Boundary $BoundaryObj -BoundaryGroupName "BGroup02"
```

Related topics

[Add-CMBoundaryToGroup](#)

[Get-CMBoundary](#)

[Get-CMBoundaryGroup](#)

[New-CMBoundary](#)

[New-CMBoundaryGroup](#)

[Remove-CMBoundary](#)

[Remove-CMBoundaryGroup](#)

[Set-CMBoundaryGroup](#)

Remove-CMBoundaryGroup

Remove-CMBoundaryGroup

Removes a boundary group.

Syntax

Parameter Set: SearchByIdMandatory

```
Remove-CMBoundaryGroup -Id <String[]> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByNameMandatory

```
Remove-CMBoundaryGroup -Name <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Remove-CMBoundaryGroup -InputObject <IResultObject> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Remove-CMBoundaryGroup** cmdlet removes a boundary group from Microsoft System Center 2012 Configuration Manager.

Parameters

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Id<String[]>

Specifies an array of identifiers (IDs) for one or more boundary groups.

Aliases	GroupId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies an input object to this cmdlet. You can get the input object by using the **Get-CMBoundaryGroup** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies the name of a boundary group.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Remove a boundary group that is specified by its ID

This command removes a boundary group that is specified by its identifier. Because the *Force* parameter is not specified, you must confirm the action before it is performed.

```
PS C:\> Remove-CMBoundaryGroup -Id "16777219"
```

Example 2: Remove multiple boundary groups by using an InputObject

The first command uses the **Get-CMBoundaryGroup** to get multiple boundary groups that are specified by their names, and stores this data into the `$BoundaryObj` variable.

The second command identifies and removes the boundaries that are specified by using the input object `$BoundaryObj`. Because the *Force* parameter is not specified, you must confirm the action before it is performed.

```
PS C:\> $BoundaryObj = Get-CMBoundary -Name "BGroup01", "BGroup02", "BGroup03"  
PS C:\> Remove-CMBoundary -InputObject $BoundaryObj
```

Related topics

[Get-CMBoundaryGroup](#)

[New-CMBoundaryGroup](#)

[Set-CMBoundaryGroup](#)

Remove-CMCategory

Remove-CMCategory

Removes a configuration category in Configuration Manager.

Syntax

Parameter Set: RemoveCategoryByCategory

```
Remove-CMCategory [-CategoryType {UserCategories | BaselineCategories | DriverCategories | AppCategories | GlobalCondition | CatalogCategories} ] [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: RemoveCategoryById

```
Remove-CMCategory [-Force] [-Id <String[]> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: RemoveCategoryByName

```
Remove-CMCategory [-Force] [-Name <String[]> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Remove-CMCategory** cmdlet removes a configuration category in Microsoft System Center 2012 Configuration Manager.

Parameters

-CategoryType<CategoryType>

Specifies a category type. Valid values are:

- UserCategories
- BaselineCategories
- DriverCategories
- AppCategories
- GlobalCondition
- CatalogCategories

The acceptable values for this parameter are:

UserCategories	
----------------	--

BaselineCategories	
DriverCategories	
AppCategories	
GlobalCondition	
CatalogCategories	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Id<String[]>

Specifies an array of IDs of configuration categories.

Aliases	CategoryinstanceUniqueid
Required?	false
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String[]>

Specifies an array of names of configuration categories.

Aliases	LocalizedCategoryInstanceName
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Remove a configuration category

This command removes the category named NewLaptopDriverSet from DriversCategories without prompting you for confirmation.

```
PS C:\> Remove-CMCategory -CategoryType "DriverCategories" -Force -Name "NewLaptopDriverSet"
```

Related topics

[Get-CMCategory](#)

[New-CMCategory](#)

Remove- CMClientAuthCertificateProfileConfigurationItem

Remove-CMClientAuthCertificateProfileConfigurationItem

Removes a certificate profile.

Syntax

Parameter Set: SearchByNameMandatory

```
Remove-CMClientAuthCertificateProfileConfigurationItem -Name <String[]> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Remove-CMClientAuthCertificateProfileConfigurationItem -Id <String[]> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Remove-CMClientAuthCertificateProfileConfigurationItem -InputObject <IResultObject> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Remove-CMClientAuthCertificateProfileConfigurationItem** cmdlet removes a certificate profile. Client computers use certificate profiles to authenticate when they use services such as a virtual private network (VPN) or a wireless network.

Parameters

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Id<String[]>

Specifies an array of IDs of certificate profiles.

Aliases	CId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a certificate profile object. To obtain a certificate profile object, use the **Get-
CMClientAuthCertificateProfileConfigurationItem** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String[]>

Specifies an array of names of certificate profiles.

Aliases	LocalizedDisplayName
Required?	true

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

[Copy-CMClientAuthCertificateProfileConfigurationItem](#)

[Get-CMClientAuthCertificateProfileConfigurationItem](#)

[New-CMClientAuthCertificateProfileConfigurationItem](#)

[Set-CMClientAuthCertificateProfileConfigurationItem](#)

Remove-CMClientOperation

Remove-CMClientOperation

Removes a Configuration Manager client operation object.

Syntax

Parameter Set: RemoveById

```
Remove-CMClientOperation -Id <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Remove-CMClientOperation** cmdlet removes a Microsoft System Center 2012 Configuration Manager client operation object.

Parameters

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Id<String>

Specifies the ID of a client operation.

Aliases	none
---------	------

Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Remove a client operation

This command removes the client operation that has the ID CMCO217. This command uses the *Force* parameter, so it does not prompt you for confirmation.

```
PS C:\> Remove-CMClientOperation -Id "CMCO217" -Force
```

Related topics

[Clear-CMClientOperation](#)

[Invoke-CMClientOperationSummarization](#)

Remove-CMClientSetting

Remove-CMClientSetting

Removes client settings.

Syntax

Parameter Set: SearchByIdMandatory

```
Remove-CMClientSetting -Id <String[]> [-Force] [-SecuredScopeNames <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByNameMandatory

```
Remove-CMClientSetting -Name <String> [-Force] [-SecuredScopeNames <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Remove-CMClientSetting -InputObject <IResultObject> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Remove-CMClientSetting** cmdlet removes a customized collection of client settings. For more information, see [About Client Settings in Configuration Manager](http://go.microsoft.com/fwlink/?LinkId=266226) (<http://go.microsoft.com/fwlink/?LinkId=266226>) on TechNet.

Parameters

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Id<String[]>

Specifies an array of identifiers for one or more collections of client settings.

Aliases	SettingsId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies an input object for this cmdlet. You can get an input object by using **Get-CMClientSetting**.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies a name for customized client settings.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SecuredScopeNames<String>

Specifies the name of security scopes. A security scope can be Default or the name of a custom-created security scope.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Remove a collection of client settings that is specified by its ID

This command removes a collection of client settings that is specified by the ID 16777255. You must confirm the action before it is performed.

```
PS C:\> Remove-CMClientSetting -Id "16777255"
```

Related topics

[Get-CMClientSetting](#)

[New-CMClientSetting](#)

[Set-CMClientSetting](#)

Remove-CMCloudDistributionPoint

Remove-CMCloudDistributionPoint

Removes cloud-based distribution points.

Syntax

Parameter Set: SearchByIdMandatory

```
Remove-CMCloudDistributionPoint -Id <String[]> [-Force] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByNameMandatory

```
Remove-CMCloudDistributionPoint -Name <String> [-Force] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Remove-CMCloudDistributionPoint -InputObject <IResultObject> [-Force] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Detailed Description

The **Remove-CMCloudDistributionPoint** cmdlet removes specified cloud-based distribution points.

When you remove a distribution point, System Center 2012 Configuration Manager deletes all the content stored there. If you want to suspend a distribution point temporarily, use the **Stop-CMCloudDistributionPoint** cmdlet.

Parameters

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-Id<String[]>

Specifies an array of identifiers for cloud distribution points. You can use a comma separated list.

Aliases	AzureServiceId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a cloud distribution point object. To obtain a cloud distribution point object, you can use the **Get-CMCloudDistributionPoint** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies the name of a cloud distribution point.

Aliases	none
Required?	true
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Remove all distribution points

This command removes all the cloud distribution points from System Center 2012 Configuration Manager. Unless you use the *Force* parameter, the cmdlet prompts you for confirmation.

```
PS C:\> Remove-CMCloudDistributionPoint
```

Example 2: Remove a distribution point using a name

This command removes the cloud distribution point named West01. Unless you use the *Force* parameter, the cmdlet prompts you for confirmation.

```
PS C:\> Remove-CMCloudDistributionPoint -Name "West01"
```

Example 3: Remove a distribution point using an ID

This command removes the cloud distribution point that has the specified identifier. Unless you use the *Force* parameter, the cmdlet prompts you for confirmation.

```
PS C:\> Remove-CMCloudDistributionPoint -Id "16777236"
```

Related topics

[Get-CMCloudDistributionPoint](#)

[New-CMCloudDistributionPoint](#)

[Set-CMCloudDistributionPoint](#)

[Start-CMCloudDistributionPoint](#)

[Stop-CMCloudDistributionPoint](#)

Remove-CMComputerAssociation

Remove-CMComputerAssociation

Deletes a computer association from Configuration Manager.

Syntax

Parameter Set: SearchByNameMandatory

```
Remove-CMComputerAssociation -DestinationComputer <String> -SourceComputer <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Remove-CMComputerAssociation -MigrationId <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Remove-CMComputerAssociation -InputObject <IResultObject> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Remove-CMComputerAssociation** cmdlet deletes a computer association from Microsoft System Center 2012 Configuration Manager. You can specify the association to remove by specifying both computers in the association or by specifying the association ID, or you can use the **Get-CMComputerAssociation** cmdlet to get an association to remove.

Parameters

-DestinationComputer<String>

Specifies the name of a destination computer.

Aliases	RestoreName
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a computer association object. To obtain a computer association object, use the **Get-CMComputerAssociation** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-MigrationId<String>

Specifies the ID of a computer association.

Aliases	none
Required?	true
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SourceComputer<String>

Specifies the name of the source computer.

Aliases	SourceName
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Remove an association by using computer names

This command removes the computer association between the computers named West155 and West073.

```
PS C:\> Remove-CMComputerAssociation -DestinationComputer "West155" -SourceComputer "West073"
```

Example 2: Remove an association by using an ID

This command removes the computer association that has the ID MID1207. This command uses the *Force* parameter, so the cmdlet does not prompt you for confirmation before it removes the association.

```
PS C:\> Remove-CMComputerAssociation -MigrationId "MID1207" -Force
```

Example 3: Remove an association by using a variable

The first command gets the computer association that has the ID MID1207, and saves it in the \$CMCA variable.

The second command removes the association saved in the \$CMCA variable. This command uses the *Force* parameter, so the cmdlet does not prompt you for confirmation before it removes the association.

```
PS C:\> $CMCA = Get-CMComputerAssociation -MigrationId "MID1207"  
PS C:\> Remove-CMComputerAssociation -InputObject $CMCA -Force
```

Related topics

[Get-CMComputerAssociation](#)

[New-CMComputerAssociation](#)

[Set-CMComputerAssociation](#)

Remove-CMConfigurationItem

Remove-CMConfigurationItem

Removes configuration items from Configuration Manager.

Syntax

Parameter Set: SearchByIdMandatory

```
Remove-CMConfigurationItem -Id <String[]> [-Force] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByNameMandatory

```
Remove-CMConfigurationItem -Name <String[]> [-Force] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Remove-CMConfigurationItem -InputObject <IResultObject> [-Force] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Detailed Description

The **Remove-CMConfigurationItem** cmdlet removes specified configuration items from Microsoft System Center 2012 Configuration Manager. You can specify items by ID, name, or by use of the **Get-CMConfigurationItem** cmdlet.

Configuration items contain one or more settings, along with compliance rules. Items usually define a unit of configuration you want to. For more information about configuration items, see [Introduction to Compliance Settings in Configuration Manager](http://go.microsoft.com/fwlink/?LinkId=211014) (<http://go.microsoft.com/fwlink/?LinkId=211014>) on TechNet.

Parameters

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Id<String[]>

Specifies an array of identifiers for one or more configuration items. You can use a comma separated list.

Aliases	CId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a configuration item object. To obtain a configuration item object, you can use the **Get-
CMConfigurationItem** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String[]>

Specifies an array of names of configuration items.

Aliases	LocalizedDisplayName
---------	----------------------

Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Remove an item using an ID

This command removes a configuration item with the specified identifier.

```
PS C:\> Remove-CMConfigurationItem -Id "16777568"
```

Example 2: Remove an item using a name

This command removes a configuration item named ConfigItem76.

```
PS C:\> Remove-CMConfigurationItem -Name "ConfigItem76"
```

Example 3: Remove an item using a variable

The first command gets a configuration item with the specified identifier and stores it in the \$CIObj variable.

The second command removes the item in the \$CIObj variable.

```
PS C:\> $CIObj=Get-CMConfigurationItem -Id "16777568"
```

```
PS C:\> Remove-CMConfigurationItem -InputObject $CIObj
```

Related topics

[Export-CMConfigurationItem](#)

[Get-CMConfigurationItem](#)

[Get-CMConfigurationItemXMLDefinition](#)

[Import-CMConfigurationItem](#)

[New-CMConfigurationItem](#)

[Set-CMConfigurationItem](#)

[Get-CMConfigurationItemHistory](#)

Remove-CMContentDistribution

Remove-CMContentDistribution

Removes packages from a distribution point.

Syntax

Parameter Set: SearchByIdMandatory_Application

```
Remove-CMContentDistribution -ApplicationId <String[]> [-CollectionName <String> ] [-  
DisableDetectAssociatedContentDependencies] [-DistributionPointGroupName <String> ] [-  
DistributionPointName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory_BootImage

```
Remove-CMContentDistribution -BootImageId <String[]> [-CollectionName <String> ] [-  
DistributionPointGroupName <String> ] [-DistributionPointName <String> ] [-Confirm] [-  
WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory_DeploymentPackage

```
Remove-CMContentDistribution -DeploymentPackageId <String[]> [-CollectionName <String> ] [-  
DistributionPointGroupName <String> ] [-DistributionPointName <String> ] [-Confirm] [-  
WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory_DriverPackage

```
Remove-CMContentDistribution -DriverPackageId <String[]> [-CollectionName <String> ] [-  
DistributionPointGroupName <String> ] [-DistributionPointName <String> ] [-Confirm] [-  
WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory_OperatingSystemImage

```
Remove-CMContentDistribution -OperatingSystemImageId <String[]> [-CollectionName <String> ]  
[-DistributionPointGroupName <String> ] [-DistributionPointName <String> ] [-Confirm] [-  
WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory_OperatingSystemInstaller

```
Remove-CMContentDistribution -OperatingSystemInstallerId <String[]> [-CollectionName  
<String> ] [-DistributionPointGroupName <String> ] [-DistributionPointName <String> ] [-  
Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory_Package

```
Remove-CMContentDistribution -PackageId <String[]> [-CollectionName <String> ] [-  
DistributionPointGroupName <String> ] [-DistributionPointName <String> ] [-Confirm] [-  
WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory_TaskSequence

```
Remove-CMContentDistribution -TaskSequenceId <String[]> [-CollectionName <String> ] [-  
DistributionPointGroupName <String> ] [-DistributionPointName <String> ] [-Confirm] [-  
WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByNameMandatory_Application

Remove-CMContentDistribution -ApplicationName <String[]> [-CollectionName <String>] [-DisableDetectAssociatedContentDependencies] [-DistributionPointGroupName <String>] [-DistributionPointName <String>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SearchByNameMandatory_BootImage

Remove-CMContentDistribution -BootImageName <String[]> [-CollectionName <String>] [-DistributionPointGroupName <String>] [-DistributionPointName <String>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SearchByNameMandatory_DeploymentPackage

Remove-CMContentDistribution -DeploymentPackageName <String[]> [-CollectionName <String>] [-DistributionPointGroupName <String>] [-DistributionPointName <String>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SearchByNameMandatory_DriverPackage

Remove-CMContentDistribution -DriverPackageName <String[]> [-CollectionName <String>] [-DistributionPointGroupName <String>] [-DistributionPointName <String>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SearchByNameMandatory_OperatingSystemImage

Remove-CMContentDistribution -OperatingSystemImageName <String[]> [-CollectionName <String>] [-DistributionPointGroupName <String>] [-DistributionPointName <String>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SearchByNameMandatory_OperatingSystemInstaller

Remove-CMContentDistribution -OperatingSystemInstallerName <String[]> [-CollectionName <String>] [-DistributionPointGroupName <String>] [-DistributionPointName <String>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SearchByNameMandatory_Package

Remove-CMContentDistribution -PackageName <String[]> [-CollectionName <String>] [-DistributionPointGroupName <String>] [-DistributionPointName <String>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SearchByNameMandatory_TaskSequence

Remove-CMContentDistribution -TaskSequenceName <String[]> [-CollectionName <String>] [-DistributionPointGroupName <String>] [-DistributionPointName <String>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SearchByValueMandatory_Application

Remove-CMContentDistribution -Application <IResultObject> [-CollectionName <String>] [-DisableDetectAssociatedContentDependencies] [-DistributionPointGroupName <String>] [-DistributionPointName <String>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SearchByValueMandatory_BootImage

Remove-CMContentDistribution -BootImage <IResultObject> [-CollectionName <String>] [-DistributionPointGroupName <String>] [-DistributionPointName <String>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SearchByValueMandatory_DeploymentPackage

Remove-CMContentDistribution -DeploymentPackage <IResultObject> [-CollectionName <String>]

```
[-DistributionPointGroupName <String> ] [-DistributionPointName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory_DriverPackage

```
Remove-CMContentDistribution -DriverPackage <IResultObject> [-CollectionName <String> ] [-DistributionPointGroupName <String> ] [-DistributionPointName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory_OperatingSystemImage

```
Remove-CMContentDistribution -OperatingSystemImage <IResultObject> [-CollectionName <String> ] [-DistributionPointGroupName <String> ] [-DistributionPointName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory_OperatingSystemInstaller

```
Remove-CMContentDistribution -OperatingSystemInstaller <IResultObject> [-CollectionName <String> ] [-DistributionPointGroupName <String> ] [-DistributionPointName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory_Package

```
Remove-CMContentDistribution -Package <IResultObject> [-CollectionName <String> ] [-DistributionPointGroupName <String> ] [-DistributionPointName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory_TaskSequence

```
Remove-CMContentDistribution -TaskSequence <IResultObject> [-CollectionName <String> ] [-DistributionPointGroupName <String> ] [-DistributionPointName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Remove-CMContentDistribution** cmdlet removes one or more packages from a distribution point.

Parameters

-Application<IResultObject>

Specifies a Configuration Manager application object. To obtain a **CMApplication** object, use the **Get-CMApplication** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ApplicationId<String[]>

Specifies an array of application IDs.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ApplicationName<String[]>

Specifies an array of application names.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-BootImage<IResultObject>

Specifies a boot image object. To obtain a **CMBootImage** object, use the **Get-CMBootImage** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-BootImageId<String[]>

Specifies an array of IDs of boot images.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-BootImageName<String[]>

Specifies an array of names of boot images.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CollectionName<String>

Specifies the name of a Configuration Manager collection.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeploymentPackage<IResultObject>

Specifies a deployment package object. To obtain a **CMDeploymentPackage** object, use the **Get-
CMDeploymentPackage** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeploymentPackageId<String[]>

Specifies an array of IDs of deployment packages.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeploymentPackageName<String[]>

Specifies an array of names of deployment packages.

Aliases	none
Required?	true
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DisableDetectAssociatedContentDependencies

Indicates that Configuration Manager automatically detects associated content dependencies and adds the associated content to the distribution for applications.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DistributionPointGroupName<String>

Specifies the name of a distribution point group.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DistributionPointName<String>

Specifies the name of a distribution point that is associated with the deployment package.

Aliases	none
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DriverPackage<IResultObject>

Specifies a driver package object. To obtain a **CMDriverPackage** object, use the **Get-CMDriverPackage** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DriverPackageld<String[]>

Specifies an array of IDs of driver packages.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DriverPackageName<String[]>

Specifies an array of names of driver packages.

Aliases	none
---------	------

Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OperatingSystemImage<IResultObject>

Specifies a **CMOperatingSystemImage** object. To obtain a **CMOperatingSystemImage** object, use the **Get-CMOperatingSystemImage** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OperatingSystemImageId<String[]>

Specifies an array of IDs of operating system images.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OperatingSystemImageName<String[]>

Specifies an array of names of operating system images.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OperatingSystemInstaller<IResultObject>

Specifies an operating system installer object. To obtain a **CMOperatingSystemInstaller** object, use the **Get-CMOperatingSystemInstaller** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OperatingSystemInstallerId<String[]>

Specifies an array of IDs of operating system installers.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OperatingSystemInstallerName<String[]>

Specifies an array of names of operating system installers.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Package<IResultObject>

Specifies a package object. To obtain a **CMPackage** object, use the **Get-CMPackage** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Packageld<String[]>

Specifies an array of IDs of packages.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PackageName<String[]>

Specifies an array of names of packages.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-TaskSequence<IResultObject>

Specifies a task sequence object. To obtain a **CMTaskSequence** object, use the **Get-CMTaskSequence** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-TaskSequenceId<String[]>

Specifies an array of IDs of task sequences.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-TaskSequenceName<String[]>

Specifies an array of names of task sequences.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

[Start-CMContentDistribution](#)

[Invoke-CMContentValidation](#)

[Get-CMApplication](#)

[Get-CMBootImage](#)

[Get-CMDeploymentPackage](#)

[Get-CMDriverPackage](#)

[Get-CMOperatingSystemImage](#)

[Get-CMOperatingSystemInstaller](#)

[Get-CMPackage](#)

[Get-CMTaskSequence](#)

Remove-CMDeployment

Remove-CMDeployment

Removes a Configuration Manager application deployment.

Syntax

Parameter Set: SearchByNameMandatory

```
Remove-CMDeployment -ApplicationName <String> -CollectionName <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Remove-CMDeployment -ApplicationName <String> -DeploymentId <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Remove-CMDeployment** cmdlet removes a Microsoft System Center 2012 Configuration Manager application deployment.

When you remove an application deployment, System Center 2012 Configuration Manager does not remove instances of the application that it has already installed. To remove these applications, you must deploy the application to computers with the action Uninstall. If you delete an application deployment, or remove a resource from the collection you are deploying to, the application will no longer be visible in Software Center or the Application Catalog.

Parameters

-ApplicationName<String>

Specifies the name of the application associated to the deployment.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-CollectionName<String>

Specifies the name of a Configuration Manager collection to which the deployment is applied.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeploymentId<String>

Specifies a deployment ID.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

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

Example 1: Remove an application deployment

This command removes the Configuration Manager deployment that is associated with the application named CMappD01 and that is applied to the collection named All Users.

```
PS C:\> Remove-CMDeployment -ApplicationName "CMappD01" -CollectionName "All Users"
```

Related topics

[Get-CMDeployment](#)

[Get-CMDeploymentType](#)

[Remove-CMDeploymentType](#)

Remove-CMDeploymentType

Remove-CMDeploymentType

Removes a deployment type.

Syntax

Parameter Set: SearchByIdMandatory

```
Remove-CMDeploymentType -ApplicationName <String> -DeploymentTypeId <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByNameMandatory

```
Remove-CMDeploymentType -ApplicationName <String> -DeploymentTypeName <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Remove-CMDeploymentType** cmdlet removes a deployment type in Microsoft System Center 2012 Configuration Manager. You cannot remove a deployment type if it is referenced by a deployment type in another application.

To remove a deployment type, you must remove any dependencies to the deployment type in other deployment types. Additionally, you must remove previous revisions of any application that contains a deployment type that references the deployment type that you want to remove. If you have already deployed the application, you cannot remove the last deployment type that the application contains, and the application must be in an active state.

Parameters

-ApplicationName<String>

Specifies the name of an application that is associated to the deployment type.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-DeploymentTypeId<String>

Specifies the ID of a deployment type.

Aliases	CIId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeploymentTypeName<String>

Specifies the name of a deployment type.

Aliases	LocalizedDisplayName
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

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

Example 1: Remove a deployment type

This command removes the deployment type named InterDept - Windows app package (.appx file) that is contained in the application named CenterApp.

```
PS C:\> Remove-CMDeploymentType -ApplicationName "CenterApp" -DeploymentTypeName "InterDept - Windows app package (.appx file)"
```

Related topics

[Add-CMDeploymentType](#)

[Get-CMDeploymentType](#)

[Set-CMDeploymentType](#)

[Get-CMDeployment](#)

Remove-CMDevice

Remove-CMDevice

Removes Configuration Manager client devices.

Syntax

Parameter Set: SearchByIdMandatory

```
Remove-CMDevice -DeviceId <String[]> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByNameMandatory

```
Remove-CMDevice -DeviceName <String[]> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Remove-CMDevice -InputObject <IResultObject> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Remove-CMDevice** cmdlet removes one or more Microsoft System Center 2012 Configuration Manager client devices. Do not remove a client if you want to uninstall the client or remove it from a collection.

Removing a device client manually deletes the client record from the System Center 2012 Configuration Manager database. Typically, you should not use this action unless it is for troubleshooting scenarios. If you delete the client record and the System Center 2012 Configuration Manager client is still installed and communicating with System Center 2012 Configuration Manager, Heartbeat Discovery recreates the client record the System Center 2012 Configuration Manager database, although the client history and any previous associations are lost.

When you remove a mobile device client that was enrolled by System Center 2012 Configuration Manager, this action also revokes the public key infrastructure (PKI) certificate that was issued to the mobile device and this certificate is then rejected by the management point, even if IIS does not check the certificate revocation list (CRL). Certificates on mobile device legacy clients are not revoked when you delete these clients.

Parameters

-DeviceId<String[]>

Specifies an array of device IDs.

Aliases	ResourceId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeviceName<String[]>

Specifies an array of device names.

Aliases	Name
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a **CMDevice** object. To obtain a **CMDevice** object, use the **Get-CMDevice** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Remove a device by using an ID

This command removes the device that has the ID 2097152000.

```
PS C:\> Remove-CMDevice -DeviceId "2097152000"
```

Example 2: Remove a device by using a name

This command removes the device named Cmcen-dist02.

```
PS C:\> Remove-CMDevice -DeviceName "Cmcen-dist02"
```

Related topics

[Get-CMDevice](#)

[Approve-CMDevice](#)

[Block-CMDevice](#)

[Unblock-CMDevice](#)

Remove-CMDeviceAffinityFromUser

Remove-CMDeviceAffinityFromUser

Removes device affinity from a Configuration Manager user.

Syntax

Parameter Set: RemoveDeviceAffinityByUserName

```
Remove-CMDeviceAffinityFromUser -UserName <String[]> [-DeviceId <String> ] [-DeviceName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: RemoveDeviceAffinityById

```
Remove-CMDeviceAffinityFromUser -UserId <String> [-DeviceId <String> ] [-DeviceName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Remove-CMDeviceAffinityFromUser** cmdlet removes device affinity from a user of Microsoft System Center 2012 Configuration Manager.

Device affinity in System Center 2012 Configuration Manager associates a user with one or more devices. Instead of deploying applications to all the devices of a user, you deploy the application to the user and System Center 2012 Configuration Manager automatically installs the application on all devices that are associated with that user. Device affinity removes the need for System Center 2012 Configuration Manager to determine the names of all the devices of a user before you deploy applications for that user.

For more information about user device affinity, see [How to Manage User Device Affinity in Configuration Manager](http://go.microsoft.com/fwlink/?linkid=247182) (<http://go.microsoft.com/fwlink/?linkid=247182>) on TechNet.

Parameters

-DeviceId<String>

Specifies a device by using an ID.

Aliases	none
Required?	false
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeviceName<String>

Specifies a device by using a name.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UserId<String>

Specifies a user by using an ID.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UserName<String[]>

Specifies an array of user names.

Aliases	UniqueUserName
Required?	true
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Remove device affinity from a user by specifying a user ID

This command adds affinity to the device named WestDivUpdates05 for the user named Patti Fuller.

```
PS C:\> Remove-CMDeviceAffinityFromUser -UserName "Patti Fuller" -DeviceName  
"WestDivUpdates05"
```

Related topics

[Add-CMDeviceAffinityToUser](#)

[Approve-CMUserDeviceAffinityRequest](#)

[Deny-CMUserDeviceAffinityRequest](#)

[Get-CMUserDeviceAffinity](#)

[Get-CMUserDeviceAffinityRequest](#)

[Import-CMUserDeviceAffinity](#)

Remove-CMDeviceCollection

Remove-CMDeviceCollection

Removes device collections from the Configuration Manager hierarchy.

Syntax

Parameter Set: SearchByIdMandatory

```
Remove-CMDeviceCollection -CollectionId <String> [-Force] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByNameMandatory

```
Remove-CMDeviceCollection -Name <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Remove-CMDeviceCollection -InputObject <IResultObject> [-Force] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Detailed Description

The **Remove-CMDeviceCollection** cmdlet removes device collections from the Microsoft System Center 2012 Configuration Manager hierarchy. You can specify the device collections by their name, ID, or an object that represents the collections.

For more information about collection, see [Introduction to Collections in Configuration Manager](http://go.microsoft.com/fwlink/p/?LinkID=259433) (<http://go.microsoft.com/fwlink/p/?LinkID=259433>) on TechNet.

Parameters

-CollectionId<String>

Specifies the IDs of the device collections to remove.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies an object that represents the device collection to remove.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies the names of the collections to remove.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

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

Example 1: Remove a device collection

This command removes the device collection named Windows® 7. You must confirm the action before the command performs it.

```
PS C:\> Remove-CMDeviceCollection -Name "Windows 7"
```

Related topics

[Export-CMDeviceCollection](#)

[Get-CMDeviceCollection](#)

[Import-CMDeviceCollection](#)

[New-CMDeviceCollection](#)

[Set-CMDeviceCollection](#)

Remove- CMDeviceCollectionDirectMembershipRule

Remove-**CMDeviceCollectionDirectMembershipRule**

Removes a direct membership rule from one or more device collections in the Configuration Manager hierarchy.

Syntax

Parameter Set: ByCollectionNameAndResourceName

```
Remove-CMDeviceCollectionDirectMembershipRule -CollectionName <String> -ResourceName <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByCollectionIdAndResourceId

```
Remove-CMDeviceCollectionDirectMembershipRule -CollectionId <String> -ResourceId <Int32> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByCollectionIdAndResourceName

```
Remove-CMDeviceCollectionDirectMembershipRule -CollectionId <String> -ResourceName <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByCollectionNameAndResourceId

```
Remove-CMDeviceCollectionDirectMembershipRule -CollectionName <String> -ResourceId <Int32> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByCollectionValueAndResourceId

```
Remove-CMDeviceCollectionDirectMembershipRule -Collection <IResultObject> -ResourceId <Int32> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByCollectionValueAndResourceName

```
Remove-CMDeviceCollectionDirectMembershipRule -Collection <IResultObject> -ResourceName <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Remove-**CMDeviceCollectionDirectMembershipRule**** cmdlet removes a direct rule from the specified collections. You can specify the collections by using their names, IDs, or by specifying an input object that represents the collections.

For more information on collection rules in Microsoft System Center 2012 Configuration Manager, see [Introduction to Collections in Configuration Manager](http://go.microsoft.com/fwlink/p/?LinkID=259433) (<http://go.microsoft.com/fwlink/p/?LinkID=259433>) on TechNet.

Parameters

-Collection<IResultObject>

Specifies a Configuration Manager device collection object. To obtain a device collection object, use the **Get-CMDeviceCollection** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CollectionId<String>

Specifies the ID of the device collection where the rule is applied.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CollectionName<String>

Specifies the name of the device collection where the rule is applied.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ResourceId<Int32>

Specifies the ID of the direct rule to remove from the device collections.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ResourceName<String>

Specifies the name of a resource in a direct membership rule.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

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

Example 1: Remove all direct membership rules from a device collection

This command removes all the direct membership rules of the device collection that has the ID CM0001A.

```
PS C:\> Remove-CMDeviceCollectionDirectMembershipRule -CollectionID "CM0001A" -ResourceId "Res_94412512"
```

Related topics

[Get-CMDeviceCollectionDirectMembershipRule](#)

[Add-CMDeviceCollectionDirectMembershipRule](#)

[Remove-CMDeviceCollectionExcludeMembershipRule](#)

[Remove-CMDeviceCollectionIncludeMembershipRule](#)

[Remove-CMDeviceCollectionQueryMembershipRule](#)

Remove- CMDeviceCollectionExcludeMembershipRule

Remove-**CMDeviceCollectionExcludeMembershipRule**

Removes an exclude membership rule from one or more device collection in the Configuration Manager hierarchy.

Syntax

Parameter Set: ByCollectionNameAndExcludeCollectionName

```
Remove-CMDeviceCollectionExcludeMembershipRule -CollectionName <String> -  
ExcludeCollectionName <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByCollectionIdAndExcludeCollectionId

```
Remove-CMDeviceCollectionExcludeMembershipRule -CollectionId <String> -ExcludeCollectionId  
<String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByCollectionIdAndExcludeCollectionName

```
Remove-CMDeviceCollectionExcludeMembershipRule -CollectionId <String> -ExcludeCollectionName  
<String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByCollectionNameAndExcludeCollectionId

```
Remove-CMDeviceCollectionExcludeMembershipRule -CollectionName <String> -ExcludeCollectionId  
<String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByCollectionValueAndExcludeCollectionId

```
Remove-CMDeviceCollectionExcludeMembershipRule -Collection <IResultObject> -  
ExcludeCollectionId <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByCollectionValueAndExcludeCollectionName

```
Remove-CMDeviceCollectionExcludeMembershipRule -Collection <IResultObject> -  
ExcludeCollectionName <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Remove-**CMDeviceCollectionExcludeMembershipRule**** cmdlet removes an exclude rule from the specified collections. You can specify the device collections by using their names, IDs, or by specifying an input object that represents the collections.

For more information about collection rules in Microsoft System Center 2012 Configuration Manager, see [Introduction to Collections in Configuration Manager](http://go.microsoft.com/fwlink/p/?LinkID=259433) (<http://go.microsoft.com/fwlink/p/?LinkID=259433>) on TechNet.

Parameters

-Collection<IResultObject>

Specifies a Configuration Manager device collection object. To obtain a device collection object, use the **Get-CMDeviceCollection** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CollectionId<String>

Specifies the ID of the device collection where the rule is applied.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CollectionName<String>

Specifies the name of the device collection where the rule is applied.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-ExcludeCollectionId<String>

Specifies the ID of the collection whose members are excluded in the device collection.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ExcludeCollectionName<String>

Specifies the name of the collection whose members are excluded from the device collections.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

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

Example 1: Remove the exclude membership rules from a device collection

This command removes the rules that exclude the members of the collection that has the ID SMSDM001 from the device collection that has the ID 9990000D.

```
PS C:\> Remove-CMUserCollectionExcludeMembershipRule -CollectionId "9990000D" -
ExcludeCollectionId "SMSDM001"
```

Related topics

[Get-CMUserCollectionExcludeMembershipRule](#)

[Add-CMUserCollectionExcludeMembershipRule](#)

Remove- CMDeviceCollectionFromAdministrativeUser

Remove-CMDeviceCollectionFromAdministrativeUser

Removes the association between an administrative user and a device collection.

Syntax

Parameter Set: RemoveDeviceCollectionFromAdminByName_Name

```
Remove-CMDeviceCollectionFromAdministrativeUser -AdministrativeUserName <String> -  
DeviceCollectionName <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: RemoveDeviceCollectionFromAdminById_Id

```
Remove-CMDeviceCollectionFromAdministrativeUser -AdministrativeUserId <Int32> -  
DeviceCollectionId <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: RemoveDeviceCollectionFromAdminById_Name

```
Remove-CMDeviceCollectionFromAdministrativeUser -AdministrativeUserName <String> -  
DeviceCollectionId <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: RemoveDeviceCollectionFromAdminById_Object

```
Remove-CMDeviceCollectionFromAdministrativeUser -AdministrativeUser <IResultObject> -  
DeviceCollectionId <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: RemoveDeviceCollectionFromAdminByName_Id

```
Remove-CMDeviceCollectionFromAdministrativeUser -AdministrativeUserId <Int32> -  
DeviceCollectionName <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: RemoveDeviceCollectionFromAdminByName_Object

```
Remove-CMDeviceCollectionFromAdministrativeUser -AdministrativeUser <IResultObject> -  
DeviceCollectionName <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: RemoveDeviceCollectionFromAdminByObject_Id

```
Remove-CMDeviceCollectionFromAdministrativeUser -AdministrativeUserId <Int32> -  
DeviceCollection <IResultObject> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: RemoveDeviceCollectionFromAdminByObject_Name

```
Remove-CMDeviceCollectionFromAdministrativeUser -AdministrativeUserName <String> -  
DeviceCollection <IResultObject> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: RemoveDeviceCollectionFromAdminByObject_Object

```
Remove-CMDeviceCollectionFromAdministrativeUser -AdministrativeUser <IResultObject> -  
DeviceCollection <IResultObject> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Remove-CMDeviceCollectionFromAdministrativeUser** cmdlet removes the association between an administrative user and a device collection. After you remove the association, you cannot automatically install an application on all device collections that are associated with that administrative user.

Parameters

-AdministrativeUser<IResultObject>

Specifies a **CMAdministrativeUser** object. To obtain a **CMAdministrativeUser** object, use the [Get-CMAdministrativeUser](#) cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AdministrativeUserId<Int32>

Specifies an ID of an administrative user.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AdministrativeUserName<String>

Specifies a name of an administrative user.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeviceCollection<IResultObject>

Specifies a **CMDeviceCollection** object. To obtain a **CMDeviceCollection** object, use the [Get-
CMDeviceCollection](#) cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeviceCollectionId<String>

Specifies the ID of a device collection.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeviceCollectionName<String>

Specifies the name of a device collection.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Remove a device collection from an administrative user

This command removes the association between the administrative user Team04\TeamAdmin and the device collection named PhoneCollection05.

```
PS C:\> Remove-CMDeviceCollectionFromAdministrativeUser -AdministrativeUserName  
"Team04\TeamAdmin" -DeviceCollectionName "PhoneCollection05"
```

Related topics

[Add-CMDeviceCollectionToAdministrativeUser](#)

[Get-CMDeviceCollection](#)

[New-CMDeviceCollection](#)

[Export-CMDeviceCollection](#)

[Import-CMDeviceCollection](#)

[Remove-CMDeviceCollection](#)

Remove- CMDeviceCollectionFromDistributionPointGroup

Remove-**CMDeviceCollectionFromDistributionPointGroup**

Removes the association between a device collection and a distribution point group.

Syntax

Parameter Set: SearchByIdMandatory

```
Remove-CMDeviceCollectionFromDistributionPointGroup -DeviceCollectionName <String> -  
DistributionPointGroupId <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByNameMandatory

```
Remove-CMDeviceCollectionFromDistributionPointGroup -DeviceCollectionName <String> -  
DistributionPointGroupName <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Remove-CMDeviceCollectionFromDistributionPointGroup -DeviceCollectionName <String> -  
InputObject <IResultObject> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Remove-**CMDeviceCollectionFromDistributionPointGroup**** cmdlet removes the association between a device collection and a distribution point group. Distribution point groups provide a logical grouping of distribution points and collections for content distribution.

Parameters

-DeviceCollectionName<String>

Specifies the device collection by using a name.

Aliases	none
Required?	true
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DistributionPointGroupId<String>

Specifies the distribution point group by using an ID.

Aliases	GroupId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DistributionPointGroupName<String>

Specifies the distribution point group by using a name.

Aliases	GroupName
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies the distribution point group by using an object that contains a distribution point group. To obtain such an object, use the [Get-CMDistributionPointGroup](#) cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Remove the association between a device collection and a distribution point group by using names

This command removes the association between the device collection named PhoneCollection05 and the distribution point group named DPG05ContosoWest.

```
PS C:\> Remove-CMDeviceCollectionFromDistributionPointGroup -DeviceCollectionName  
"PhoneCollection05" - DistributionPointGroupName "DPG05ContosoWest"
```

Related topics

[Add-CMDeviceCollectionToDistributionPointGroup](#)

[Get-CMDistributionPointGroup](#)

Remove- CMDeviceCollectionIncludeMembershipRule

Remove-**CMDeviceCollectionIncludeMembershipRule**

Removes an include membership rule from one or more device collection in the Configuration Manager hierarchy.

Syntax

Parameter Set: ByCollectionNameAndIncludeCollectionName

```
Remove-CMDeviceCollectionIncludeMembershipRule -CollectionName <String> -  
IncludeCollectionName <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByCollectionIdAndIncludeCollectionId

```
Remove-CMDeviceCollectionIncludeMembershipRule -CollectionId <String> -IncludeCollectionId  
<String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByCollectionIdAndIncludeCollectionName

```
Remove-CMDeviceCollectionIncludeMembershipRule -CollectionId <String> -IncludeCollectionName  
<String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByCollectionNameAndIncludeCollectionId

```
Remove-CMDeviceCollectionIncludeMembershipRule -CollectionName <String> -IncludeCollectionId  
<String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByCollectionValueAndIncludeCollectionId

```
Remove-CMDeviceCollectionIncludeMembershipRule -Collection <IResultObject> -  
IncludeCollectionId <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByCollectionValueAndIncludeCollectionName

```
Remove-CMDeviceCollectionIncludeMembershipRule -Collection <IResultObject> -  
IncludeCollectionName <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Remove-**CMDeviceCollectionIncludeMembershipRule**** cmdlet removes an include rule from the specified collections. You can specify the device collections by using their names, Ids, or by specifying an input object that represents the collections.

For more information about collection rules, see [Introduction to Collections in Configuration Manager](http://go.microsoft.com/fwlink/p/?LinkID=259433) (<http://go.microsoft.com/fwlink/p/?LinkID=259433>) on TechNet.

Parameters

-Collection<IResultObject>

Specifies a Configuration Manager device collection object. To obtain a device collection object, use the **Get-CMDeviceCollection** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CollectionId<String>

Specifies the ID of the device collection where the rule is applied.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CollectionName<String>

Specifies the name of the device collection where the rule is applied.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-IncludeCollectionId<String>

Specifies the ID of the collection whose members are included in the device collection.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-IncludeCollectionName<String>

Specifies the name of the collection whose members are included in the device collections.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

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

Example 1: Removes an include membership rule from a single collection

This command removes the rule for the All Mobile Devices, which has the ID SMSD001, collection from the device collection named Windows 7.

```
PS C:\> Remove-CMDeviceCollectionIncludeMembershipRule -Name "Windows 7" -  
IncludecollectionId "SMSD001"
```

Related topics

[Add-CMDeviceCollectionIncludeMembershipRule](#)

[Get-CMDeviceCollectionIncludeMembershipRule](#)

[Get-CMDeviceCollection](#)

[Remove-CMDeviceCollectionExcludeMembershipRule](#)

[Remove-CMDeviceCollectionQueryMembershipRule](#)

Remove- CMDeviceCollectionQueryMembershipRule

Remove-**CMDeviceCollectionQueryMembershipRule**

Removes a query membership rule from one or more device collection in the Configuration Manager hierarchy.

Syntax

Parameter Set: ByCollectionName

```
Remove-CMDeviceCollectionQueryMembershipRule -CollectionName <String> -RuleName <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByCollectionId

```
Remove-CMDeviceCollectionQueryMembershipRule -CollectionId <String> -RuleName <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByCollectionValue

```
Remove-CMDeviceCollectionQueryMembershipRule -Collection <IResultObject> -RuleName <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Remove-**CMDeviceCollectionQueryMembershipRule**** cmdlet removes a query rule from the specified device collections. You can specify the device collections by using their names, Ids, or by specifying an input object that represents the collections.

For more information about membership rules, see [Introduction to Collections in Configuration Manager](http://go.microsoft.com/fwlink/p/?LinkID=259433) (<http://go.microsoft.com/fwlink/p/?LinkID=259433>) on TechNet.

Parameters

-Collection<IResultObject>

Specifies a Configuration Manager device collection object. To obtain a device collection object, use the **Get-**CMDeviceCollection**** cmdlet.

Aliases	none
Required?	true

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CollectionId<String>

Specifies the IDs of the device collections.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CollectionName<String>

Specifies the name of the device collection where the rule is applied.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RuleName<String>

Specifies the name of the query rule to remove from the device collections.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Remove the query membership rules for a device collection

This command removes the query membership rule named TPM Information from the device collection named Mobile Windows 7 Devices.

```
PS C:\> Remove-CMDeviceCollectionQueryMembershipRule -CollectionName "Mobile Windows 7 Devices" -RuleName "TPM Information"
```

Related topics

[Get-CMDeviceCollectionQueryMembershipRule](#)

[Add-CMDeviceCollectionQueryMembershipRule](#)

[Get-CMDeviceCollection](#)

Remove-CMDeviceCollectionVariable

Remove-CMDeviceCollectionVariable

Removes a task sequence variable that is associated with a device collection in the Configuration Manager hierarchy.

Syntax

Parameter Set: SearchByIdMandatory

```
Remove-CMDeviceCollectionVariable -CollectionId <String> -VariableName <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByNameMandatory

```
Remove-CMDeviceCollectionVariable -CollectionName <String> -VariableName <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Remove-CMDeviceCollectionVariable -Collection <IResultObject> -VariableName <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Remove-CMDeviceCollectionVariable** cmdlet removes a task sequence variable that is associated with the specified device collection in Microsoft System Center 2012 Configuration Manager. These variables are used by task sequences that are deployed to the collection.

Parameters

-Collection<IResultObject>

Specifies an object that represents the device collection associated with variable.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-CollectionId<String>

Specifies the ID of a device collection.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CollectionName<String>

Specifies the name of a device collection.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-VariableName<String>

Specifies the name of the task sequence variable to remove.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Remove a task sequence variable from a device collection

This command removes a task sequence variable named New_ComputerName that is associated with a device collection named All Unknown Devices.

```
PS C:\> Remove-CMDeviceCollectionVariable -CollectionName "All Unknown Devices" -VariableName "New_ComputerName"
```

Related topics

[Get-CMDeviceCollectionVariable](#)

[Set-CMDeviceCollectionVariable](#)

[New-CMDeviceCollectionVariable](#)

Remove-CMDeviceVariable

Remove-CMDeviceVariable

Removes a variable defined for a device.

Syntax

Parameter Set: SearchByIdMandatory

```
Remove-CMDeviceVariable -ResourceId <String> -VariableName <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByNameMandatory

```
Remove-CMDeviceVariable -DeviceName <String> -VariableName <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Remove-CMDeviceVariable -Device <IResultObject> -VariableName <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Remove-CMDeviceVariable** cmdlet removes a variable defined for a device.

Parameters

-Device<IResultObject>

Specifies a **CMDevice** object. To obtain a **CMDevice** object, use the **Get-CMDevice** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeviceName<String>

Specifies a device name. You can specify a NetBIOS name or a fully qualified domain name (FQDN).

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ResourceId<String>

Specifies a Systems Management Server (SMS) ID.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-VariableName<String>

Specifies the name of the device variable.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

[Get-CMDeviceVariable](#)

[New-CMDeviceVariable](#)

[Set-CMDeviceVariable](#)

Remove-CMDistributionPoint

Remove-CMDistributionPoint

Removes a distribution point.

Syntax

Parameter Set: SearchByNameMandatory

```
Remove-CMDistributionPoint -SiteCode <String> -SiteSystemServerName <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Remove-CMDistributionPoint -InputObject <IResultObject> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Remove-CMDistributionPoint** cmdlet removes a distribution point. When you remove a distribution point, you remove the designation of a site system server to function as a distribution center for content files for applications, packages, software updates, and operating system deployment.

Parameters

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a **CMDistributionPoint** object. To obtain a **CMDistributionPoint** object, use the **Get-CMDistributionPoint** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteCode<String>

Specifies the site code that is associated with the distribution point.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteSystemServerName<String>

Specifies a fully qualified domain name (FQDN) of the server that hosts the site system role.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Remove a distribution point by using a site code and site system role

This command removes the distribution point that is associated with the site code CM1 and the site system role named CMCEN-DIST02.TSQA.CORP.CONTOSCO.COM.

```
PS C:\> Remove-CMDistributionPoint -SiteSystemServerName "CMCEN-DIST02.TSQA.CORP.CONTOSCO.COM" -SiteCode "CM1"
```

Example 2: Remove a distribution point by using an object variable

The first command gets a distribution that is associated with the site code ST1 and the site system role named PptCmdv-1.DIST01.CORP.CONTOSCO.com. The command stores the results to the \$Di variable.

The second command removes the distribution point stored in the \$Di variable.

```
PS C:\> $Di = Get-CMDistributionPoint -SiteSystemServerName "PptCmdv-1.DIST01.CORP.CONTOSCO.com" -SiteCode "ST1"
PS C:\> Remove-CMDistributionPoint -InputObject $Di
```

Related topics

[Add-CMDistributionPoint](#)

[Get-CMDistributionPoint](#)

[New-CMDistributionPointGroup](#)

[Set-CMDistributionPoint](#)

[Get-CMDistributionPointGroup](#)

[Update-CMDistributionPoint](#)

Remove-CMDistributionPointFromGroup

Remove-CMDistributionPointFromGroup

Removes a Configuration Manager distribution point from a distribution point group.

Syntax

Parameter Set: RemoveDistributionPointFromGroupById_Id

```
Remove-CMDistributionPointFromGroup -DistributionPointGroupId <String> -DistributionPointId <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: RemoveDistributionPointFromGroupById_Name

```
Remove-CMDistributionPointFromGroup -DistributionPointGroupName <String> -DistributionPointId <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: RemoveDistributionPointFromGroupById_Object

```
Remove-CMDistributionPointFromGroup -DistributionPointGroup <IResultObject> -DistributionPointId <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: RemoveDistributionPointFromGroupByName_Id

```
Remove-CMDistributionPointFromGroup -DistributionPointGroupId <String> -DistributionPointName <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: RemoveDistributionPointFromGroupByName_Name

```
Remove-CMDistributionPointFromGroup -DistributionPointGroupName <String> -DistributionPointName <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: RemoveDistributionPointFromGroupByName_Object

```
Remove-CMDistributionPointFromGroup -DistributionPointGroup <IResultObject> -DistributionPointName <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: RemoveDistributionPointFromGroupByObject_Id

```
Remove-CMDistributionPointFromGroup -DistributionPoint <IResultObject> -DistributionPointGroupId <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: RemoveDistributionPointFromGroupByObject_Name

```
Remove-CMDistributionPointFromGroup -DistributionPoint <IResultObject> -DistributionPointGroupName <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: RemoveDistributionPointFromGroupByObject_Object

```
Remove-CMDistributionPointFromGroup -DistributionPoint <IResultObject> -DistributionPointGroup <IResultObject> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Remove-CMDistributionPointFromGroup** cmdlet removes a Microsoft System Center 2012 Configuration Manager distribution point from a distribution point group. Distribution point groups provide a logical grouping of distribution points for content distribution.

To remove a distribution point, specify both the distribution point to remove and the distribution point group. You can specify these values by using an ID or a name, or you can use the **Get-CMDistributionPoint** cmdlet or the **Get-CMDistributionPointGroup** cmdlet to obtain the relevant object.

Parameters

-DistributionPoint<IResultObject>

Specifies a distribution point object. To obtain a distribution point object, use the **Get-CMDistributionPoint** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DistributionPointGroup<IResultObject>

Specifies a distribution point group object. To obtain a distribution point group object, use the **Get-CMDistributionPointGroup** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DistributionPointGroupId<String>

Specifies the ID of a distribution point group.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DistributionPointGroupName<String>

Specifies the name of a distribution point group.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DistributionPointId<String>

Specifies the ID of a distribution point.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DistributionPointName<String>

Specifies the name of a distribution point.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Remove a distribution point by using an ID

This command removes a distribution point that has an ID of SMS000022 from a distribution point group that has the ID SMS000067.

```
PS C:\> Remove-CMDistributionPointFromGroup -DistributionPointGroupId "SMS000067" -  
DistributionPointId "SMS000022"
```

Example 2: Remove a distribution point by using a name

This command removes a distribution point, specified by its name, from a distribution point group that has the ID SMS000067. This command uses the *Force* parameter, therefore, it does not prompt you before it removes the distribution point.

```
PS C:\> Remove-CMDistributionPointFromGroup -DistributionPointGroupId "SMS000067" -  
DistributionPointName "Western office distribution point" -Force
```

Related topics

[Add-CMDistributionPointToGroup](#)

[Get-CMDistributionPoint](#)

[Get-CMDistributionPointGroup](#)

Remove-CMDistributionPointGroup

Remove-CMDistributionPointGroup

Removes distribution point groups.

Syntax

Parameter Set: SearchByIdMandatory

```
Remove-CMDistributionPointGroup -Id <String[]> [-Force] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByNameMandatory

```
Remove-CMDistributionPointGroup -Name <String> [-Force] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Remove-CMDistributionPointGroup -InputObject <IResultObject> [-Force] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Detailed Description

The **Remove-CMDistributionPointGroup** cmdlet removes one or more distribution point groups. When you remove a distribution point group, you cannot use the distribution point group to distribute content to the site collections that are associated with the distribution point group and to the distribution points that are members of the distribution point group.

Parameters

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-Id<String[]>

Specifies an array of IDs of distribution point groups.

Aliases	GroupId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a **CMDistributionPointGroup** object. To obtain a **CMDistributionPointGroup** object, use the [Get-CMDistributionPointGroup](#) cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies a name of a distribution point group.

Aliases	none
Required?	true
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
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](#).

Inputs

The input type is the type of the objects that you can pipe to the cmdlet.

- **Specifies an array of names of distribution point groups.**

Examples

Example 1: Remove a distribution point group by using an ID

This command removes the distribution point group that has the ID 03BCD6FE-5604-4725-B650-DD1EA03676DE.

```
PS C:\> Remove-CMDistributionPointGroup -Id "{03BCD6FE-5604-4725-B650-DD1EA03676DE}"
```

Example 2: Remove a distribution point group by using a name

This command removes the distribution point group named DpgDept01.

```
PS C:\> Remove-CMDistributionPointGroup -Name "DpgDept01"
```

Related topics

[New-CMDistributionPointGroup](#)

[Get-CMDistributionPointGroup](#)

[Set-CMDistributionPointGroup](#)

Remove-CMDriver

Remove-CMDriver

Removes a software driver or a device driver.

Syntax

Parameter Set: SearchByIdMandatory

```
Remove-CMDriver -Id <String[]> [-Force] [-SecuredScopeNames <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByNameMandatory

```
Remove-CMDriver -Name <String[]> [-Force] [-SecuredScopeNames <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Remove-CMDriver -InputObject <IResultObject> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Remove-CMDriver** cmdlet removes a software driver or device driver from Microsoft System Center 2012 Configuration Manager. After it is removed, a driver cannot be used in a driver package.

Parameters

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Id<String[]>

Specifies an array of identifiers for a driver.

Aliases	Ciid
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a driver object. To obtain a driver object, use the **Get-CMDriver** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String[]>

Specifies an array of names for a driver.

Aliases	LocalizedDisplayName
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SecuredScopeNames<String>

Specifies an array of secured scope names.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Remove a driver that is specified by its name

This command removes a device driver that is specified by its name.

```
PS C:\> Remove-CMDriver -Name "cdrom.sys"
```

Related topics

[Disable-CMDriver](#)

[Enable-CMDriver](#)

[Get-CMDriverPackage](#)

Remove-CMDriverFromDriverPackage

Remove-CMDriverFromDriverPackage

Removes a driver from a driver package.

Syntax

Parameter Set: RemoveDriverFromDriverPackageById_Id

```
Remove-CMDriverFromDriverPackage -DriverId <String> -DriverPackageId <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: RemoveDriverFromDriverPackageById_Name

```
Remove-CMDriverFromDriverPackage -DriverId <String> -DriverPackageName <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: RemoveDriverFromDriverPackageById_Object

```
Remove-CMDriverFromDriverPackage -DriverId <String> -DriverPackage <IResultObject> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: RemoveDriverFromDriverPackageByName_Id

```
Remove-CMDriverFromDriverPackage -DriverName <String> -DriverPackageId <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: RemoveDriverFromDriverPackageByName_Name

```
Remove-CMDriverFromDriverPackage -DriverName <String> -DriverPackageName <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: RemoveDriverFromDriverPackageByName_Object

```
Remove-CMDriverFromDriverPackage -DriverName <String> -DriverPackage <IResultObject> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: RemoveDriverFromDriverPackageByObject_Id

```
Remove-CMDriverFromDriverPackage -Driver <IResultObject> -DriverPackageId <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: RemoveDriverFromDriverPackageByObject_Name

```
Remove-CMDriverFromDriverPackage -Driver <IResultObject> -DriverPackageName <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: RemoveDriverFromDriverPackageByObject_Object

```
Remove-CMDriverFromDriverPackage -Driver <IResultObject> -DriverPackage <IResultObject> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Remove-CMDriverFromDriverPackage** cmdlet removes a driver from a driver package in Microsoft System Center 2012 Configuration Manager. When you remove a driver from a driver package, the device driver content is deleted from the source directory share for the driver package.

Parameters

-Driver<IResultObject>

Specifies a driver object. To obtain a **CMDriver** object, use the [Get-CMDriver](#) cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DriverId<String>

Specifies the ID of a driver.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DriverName<String>

Specifies the name of a driver.

Aliases	none
Required?	true

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DriverPackage<IResultObject>

Specifies a **CMDriverPackage** object. To obtain a **CMDriverPackage** object, use the [Get-CMDriverPackage](#) cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DriverPackageld<String>

Specifies the ID of a driver package.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DriverPackageName<String>

Specifies the name of a driver package.

Aliases	none
---------	------

Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Remove a driver from a driver package

This command removes the driver named Adaptec Embedded SCSI HostRAID Controller from the boot image named DrvPkg01.

```
PS C:\> Remove-CMDriverfromDriverPackage -DriverName "Adaptec Embedded SCSI HostRAID  
Controller" -DriverPackageName "DrvPkg01"
```

Related topics

[Add-CMDriverToDriverPackage](#)

[Get-CMDriver](#)

Remove-CMDriverPackage

Remove-CMDriverPackage

Removes a driver package.

Syntax

Parameter Set: SearchByIdMandatory

```
Remove-CMDriverPackage -Id <String[]> [-Force] [-SecuredScopeNames <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByNameMandatory

```
Remove-CMDriverPackage -Name <String> [-Force] [-SecuredScopeNames <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Remove-CMDriverPackage -InputObject <IResultObject> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Remove-CMDriverPackage** cmdlet removes a driver package from Microsoft System Center 2012 Configuration Manager. After the driver package is removed, it cannot be used by any task sequence.

Parameters

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Id<String[]>

Specifies an array of identifiers for a driver package.

Aliases	Packageld
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a driver package object. To obtain a driver package object, use the **Get-CMDriverPackage** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies the name of a driver.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SecuredScopeNames<String>

Specifies an array of secured scope names.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Remove a driver package that is specified by its identifier

This command removes a driver package that is specified by its identifier.

```
PS C:\> Remove-CMDriverPackage -Id "ST100062"
```

Related topics

[Export-CMDriverPackage](#)

[Get-CMDriverPackage](#)

[Import-CMDriverPackage](#)

[New-CMDriverPackage](#)

[Set-CMDriverPackage](#)

Remove-CMEndpointProtectionPoint

Remove-CMEndpointProtectionPoint

Removes an Endpoint Protection point.

Syntax

Parameter Set: SearchByNameMandatory

```
Remove-CMEndpointProtectionPoint -SiteCode <String> -SiteSystemServerName <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Remove-CMEndpointProtectionPoint -InputObject <IResultObject> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Remove-CMEndpointProtectionPoint** cmdlet removes a System Center 2012 Endpoint Protection point from Microsoft System Center 2012 Configuration Manager. For more information about Endpoint Protection in Configuration Manager, see [Endpoint Protection in Configuration Manager](http://go.microsoft.com/fwlink/?LinkId=268427) (<http://go.microsoft.com/fwlink/?LinkId=268427>) on TechNet.

Parameters

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies the input to this cmdlet. To obtain an input object, use the **Get-CMEndpointProtectionPoint** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteCode<String>

Specifies a site code.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteSystemServerName<String>

Specifies a fully qualified domain name (FQDN) of the server that hosts the site system role.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Remove an Endpoint Protection point

This command removes an Endpoint Protection point.

```
PS C:\> Remove-CMEndpointProtectionPoint -SiteSystemServerName "CMServer01.Contoso.com" -SiteCode "CM1"
```

Example 2: Remove an Endpoint Protection point by using an input object

The first command uses the **Get-CMEndpointProtectionPoint** cmdlet to get an Endpoint Protection point object and assign it to the variable \$EPP.

The second command removes the Endpoint Protection point object that is assigned to the variable \$EPP.

```
PS C:\> $EPP = Get-CMEndpointProtectionPoint -SiteCode "CM1" -SiteSystemServerName  
"CMServer01.Contoso.com"  
PS C:\> Remove-CMEndpointProtectionPoint -InputObject $EPP
```

Related topics

[Add-CMEndpointProtectionPoint](#)

[Get-CMEndpointProtectionPoint](#)

[Set-CMEndpointProtectionPoint](#)

Remove-CMEnrollmentPoint

Remove-CMEnrollmentPoint

Removes an enrollment point.

Syntax

Parameter Set: SearchByNameMandatory

```
Remove-CMEnrollmentPoint -SiteCode <String> -SiteSystemServerName <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Remove-CMEnrollmentPoint -InputObject <IResultObject> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Remove-CMEnrollmentPoint** cmdlet removes an enrollment point in Microsoft System Center 2012 Configuration Manager. An enrollment point is a site system role that uses public key infrastructure (PKI) certificates to complete mobile device enrollment and to provision Intel AMT-based computers. After you remove an enrollment point, client computers and devices must use a different enrollment point.

Parameters

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies the input to this cmdlet. You can use the **Get-CMEnrollmentPoint** cmdlet to get an input object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteCode<String>

Specifies a site code.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteSystemServerName<String>

Specifies the fully qualified domain name (FQDN) of the server that hosts the site system role.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Remove an enrollment point

This command removes an enrollment point.

```
PS C:\> Remove-CMEnrollmentPoint -SiteSystemServerName "SiteServer01.Contoso.com" -SiteCode "CM1"
```

Related topics

[Add-CMEnrollmentPoint](#)

[Get-CMEnrollmentPoint](#)

[Set-CMEnrollmentPoint](#)

Remove-CMEnrollmentProxyPoint

Remove-CMEnrollmentProxyPoint

Removes an enrollment proxy point.

Syntax

Parameter Set: SearchByNameMandatory

```
Remove-CMEnrollmentProxyPoint -SiteCode <String> -SiteSystemServerName <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Remove-CMEnrollmentProxyPoint -InputObject <IResultObject> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Remove-CMEnrollmentProxyPoint** cmdlet removes an enrollment proxy point in Microsoft System Center 2012 Configuration Manager. An enrollment proxy point is a site system role that manages enrollment requests from mobile devices. After you remove an enrollment proxy point, mobile devices must use a different enrollment proxy point.

Parameters

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies the input to this cmdlet. You can get an input object by using the **Get-CMEnrollmentProxyPoint** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteCode<String>

Specifies a site code.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteSystemServerName<String>

Specifies the fully qualified domain name (FQDN) of the server that hosts the site system role.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Remove an enrollment proxy point

This command removes an enrollment proxy point.

```
PS C:\> Remove-CMEnrollmentProxyPoint -SiteCode "CM1" -SiteSystemServerName "SiteServer01.Contoso.com"
```

Related topics

[Add-CMEnrollmentProxyPoint](#)

[Get-CMEnrollmentProxyPoint](#)



Remove-CMExchangeServer

Remove-CMExchangeServer

Removes an Exchange Server object from Configuration Manager.

Syntax

Parameter Set: SearchBySiteCodeMandatory

```
Remove-CMExchangeServer -Address <String> -SiteCode <String> [-Force] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Detailed Description

The **Remove-CMExchangeServer** cmdlet removes a Microsoft Exchange Server object from Microsoft System Center 2012 Configuration Manager for one or more System Center 2012 Configuration Manager sites. This cmdlet does not uninstall the Exchange Server.

System Center 2012 Configuration Manager works with Exchange Server to manage mobile devices that cannot run System Center 2012 Configuration Manager clients.

Parameters

-Address<String>

Specifies a URL for the Exchange Server.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteCode<String>

Specifies the site code for a Configuration Manager site associated with the Exchange Server.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Remove an Exchange Server

This command removes the Exchange Server with the specified address for the site code PE1.

```
PS C:\> Remove-CMExchangeServer -Address "http://localhost/PowerShell" -SiteCode "PE1"
```

Related topics

[Get-CMExchangeServer](#)

[New-CMExchangeServer](#)

[Set-CMExchangeServer](#)

[Sync-CMExchangeServer](#)

Remove-CMFallbackStatusPoint

Remove-CMFallbackStatusPoint

Removes a Configuration Manager fallback status point.

Syntax

Parameter Set: SearchByNameMandatory

```
Remove-CMFallbackStatusPoint -SiteCode <String> -SiteSystemServerName <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Remove-CMFallbackStatusPoint -InputObject <IResultObject> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Remove-CMFallbackStatusPoint** cmdlet removes a specified fallback status point site system role. You can specify the site system name and site code for a fallback status point or use the **Get-CMFallbackStatusPoint** cmdlet to obtain a fallback status point object.

Microsoft System Center 2012 Configuration Manager can use one or more fallback status points to collect state messages for a site and send them on to System Center 2012 Configuration Manager. After you remove a fallback status point, that system no longer forwards state messages.

The use of a fallback status point is optional. You can use this cmdlet to remove redundant fallback status points or to remove the last fallback status point from a site if you do not want to use that role.

Parameters

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-InputObject<IResultObject>

Specifies a fallback status point role. To obtain a fallback status point role, use the **Get-CMFallbackStatusPoint** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteCode<String>

Specifies the site code for a fallback status point.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteSystemServerName<String>

Specifies the site system name for a fallback status point.

Aliases	none
Required?	true
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Remove a specified fallback status point

This command removes the fallback status point for the site with the site code CM1 and the system name Server21.West01.Contoso.com.

```
PS C:\> Remove-CMFallbackStatusPoint -SiteCode "CM1" -SiteSystemServerName
"Server21.West01.Contoso.com"
```

Example 2: Remove a fallback status point object

The first command gets a fallback status point for the site with the site code CM1 and the system name Server21.West01.Contoso.com and stores that object in the \$CMFSP variable.

The second command removes the object stored in \$CMFSP.

```
PS C:\> $CMFSP = Get-CMFallbackStatusPoint -SiteCode "CM1" -SiteSystemServerName  
"Server21.West01.Contoso.com"
```

```
PS C:\> Remove-CMFallbackStatusPoint -InputObject $CMFSP
```

Related topics

[Add-CMFallbackStatusPoint](#)

[Get-CMFallbackStatusPoint](#)

[Set-CMFallbackStatusPoint](#)

Remove-CMFileReplicationRoute

Remove-CMFileReplicationRoute

Removes a file replication route from Configuration Manager.

Syntax

Parameter Set: SearchBySiteCodeMandatory

```
Remove-CMFileReplicationRoute -DestinationSiteCode <String> -SourceSiteCode <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Remove-CMFileReplicationRoute** cmdlet removes a file replication route from Microsoft System Center 2012 Configuration Manager. System Center 2012 Configuration Manager uses file replication routes to transfer file-based data between sites in a hierarchy. Each file replication route identifies a destination site to which file-based data can transfer.

File replication routes were known as addresses in versions of Configuration Manager before System Center 2012 Configuration Manager. The functionality of file replication routes is the same as that of addresses in earlier versions.

Parameters

-DestinationSiteCode<String>

Specifies the destination site code for the file replication route that you remove.

Aliases	DesSiteCode
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SourceSiteCode<String>

Specifies the source site code for the file replication route that you remove.

Aliases	SiteCode
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Remove a file replication route

This command removes a file replication route from the site that has the site code IM1 to the site that has the site code IM5.

```
PS C:\> Remove-CMFileReplicationRoute -DestinationSiteCode "IM5" -SourceSiteCode "IM1"
```

Related topics

[Get-CMFileReplicationRoute](#)

[New-CMFileReplicationRoute](#)

[Set-CMFileReplicationRoute](#)

Remove-CMGlobalCondition

Remove-CMGlobalCondition

Removes a global condition object.

Syntax

Parameter Set: SearchByIdMandatory

```
Remove-CMGlobalCondition -Id <String[]> [-Force] [-SecuredScopeNames <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByNameMandatory

```
Remove-CMGlobalCondition -Name <String[]> [-Force] [-SecuredScopeNames <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Remove-CMGlobalCondition -InputObject <IResultObject> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Remove-CMGlobalCondition** cmdlet removes a global condition object.

Microsoft System Center 2012 Configuration Manager uses global conditions to represent business or technical conditions. Global conditions specify how to provide and deploy applications to client devices.

You can specify a global condition by name or ID or use the **Get-CMGlobalCondition** cmdlet to obtain a global condition object. You cannot remove read-only global conditions.

Parameters

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Id<String[]>

Specifies an array of identifiers of global conditions. This value corresponds to the **CI_ID** property of a global condition object.

Aliases	CIId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a global condition object. To obtain a global condition object, use the **Get-CMGlobalCondition** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String[]>

Specifies an array of names for global conditions. This value corresponds to the **LocalizedDisplayName** property of a global condition object.

Aliases	LocalizedDisplayName
Required?	true

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SecuredScopeNames<String>

Specifies a security scope of a global condition. You can use this parameter to narrow your specified global conditions by the current security scope.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Remove a global condition

This command removes a global condition named GC56. Because the command uses the *Force* parameter, the system does not prompt you before it removes the condition.

```
PS C:\> Remove-CMGlobalCondition -Name "GC56" -Force
```

Example 2: Remove a global condition using a variable

The first command uses the **Get-CMGlobalCondition** cmdlet to get the global condition named GC57 and stores it in the \$CMGC variable.

The second command removes the global condition stored in that variable. This command does not use the *Force* parameter, so it prompts you for confirmation before it removes the global condition.

```
PS C:\> $CMGC = Get-CMGlobalCondition -Name "GC57"  
PS C:\> Remove-CMGlobalCondition -InputObject $CMGC
```

Remove

Are you sure you wish to remove GlobalCondition:

LocalizedDisplayName=" GC57"?

[Y] Yes [N] No [S] Suspend [?] Help (default is "Y"): Y

Related topics

[Get-CMGlobalCondition](#)

[New-CMGlobalCondition](#)

[Set-CMGlobalCondition](#)

Remove-CMHardwareRequirement

Remove-CMHardwareRequirement

Removes Configuration Manager hardware requirement objects for products.

Syntax

Parameter Set: SearchByNameMandatory

```
Remove-CMHardwareRequirement -Product <String> [-Force] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Remove-CMHardwareRequirement -InputObject <IResultObject> [-Force] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Detailed Description

The **Remove-CMHardwareRequirement** cmdlet removes hardware requirement objects from Microsoft System Center 2012 Configuration Manager.

System Center 2012 Configuration Manager manages Asset Intelligence information, including hardware requirements, for different software products. You can add, modify, or delete your own hardware requirements, but you cannot change built-in hardware requirement objects.

You can use this cmdlet to remove hardware requirement objects. You can specify a product by name or obtain a requirement by using the **Get-CMHardwareRequirement** cmdlet.

Parameters

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-InputObject<IResultObject>

Specifies a hardware requirement object. To obtain a hardware requirement object, use the **Get-CMHardwareRequirement** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Product<String>

Specifies the name of a software product name.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Remove a hardware requirement

This command removes the hardware requirement object for a product named Accounts Program.

```
PS C:\> Remove-CMHardwareRequirement -Product "Accounts Program"
```

Remove

Are you sure you wish to remove HardwareRequirement: Product="Accounts Program"?

[Y] Yes [N] No [S] Suspend [?] Help (default is "Y"):

Related topics

[Get-CMHardwareRequirement](#)

[New-CMHardwareRequirement](#)

[Set-CMHardwareRequirement](#)

Remove-CMMaintenanceWindow

Remove-CMMaintenanceWindow

Removes a maintenance window.

Syntax

Parameter Set: Default

```
Remove-CMMaintenanceWindow [-CollectionID] <String> -Name <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Remove-CMMaintenanceWindow** cmdlet removes a maintenance window associated with a collection. If you remove a maintenance window during that window, the maintenance updates finish installation.

Parameters

-CollectionID<String>

Specifies the ID of the collection that the maintenance window applies to.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

-Name<String>

Specifies the name of the maintenance window.

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](#).

Related topics

[Get-CMMaintenanceWindow](#)

[New-CMMaintenanceWindow](#)

[Set-CMMaintenanceWindow](#)

Remove-CMManagementPoint

Remove-CMManagementPoint

Removes a management point.

Syntax

Parameter Set: SearchByNameMandatory

```
Remove-CMManagementPoint -SiteCode <String> -SiteSystemServerName <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Remove-CMManagementPoint -InputObject <IResultObject> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Remove-CMManagementPoint** cmdlet removes a management point. A management point is a site system role that provides policy and service location information to clients and receives configuration data from clients.

When you remove a management point, Microsoft System Center 2012 Configuration Manager disables communication between the site server and the clients that you assigned to the site server. System Center 2012 Configuration Manager cannot provide these clients with installation prerequisites, client installation files, configuration details, advertisements, and software distribution package source file locations. Additionally, System Center 2012 Configuration Manager cannot receive inventory data, software metering information, and status and state messages from the clients.

Parameters

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a **CManagementPoint** object. To obtain a **CManagementPoint** object, use the **Get-CManagementPoint** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteCode<String>

Specifies the site code of the Configuration Manager site that hosts the site system role.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteSystemServerName<String>

Specifies the fully qualified domain name (FQDN) of the server that hosts the site system role.

Aliases	none
Required?	true
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Remove a management point

This command removes the management point from the Configuration Manager site that has the site code CM1 on the site system named cmcen-dist02.tsqa.contoso.com.

```
PS C:\> Remove-CMManagementPoint -SiteSystemServerName "cmcen-dist02.tsqa.contoso.com" -
SiteCode "CM1"
```

Example 2: Remove a management point by using an object variable

The first command gets the management point from the Configuration Manager site that has the site code CM1 on the site system named dist02.tsqa.contoso.com. The command stores the results in the \$Mp variable.

The second command removes the management point stored in the \$Mp variable.

```
PS C:\> $Mp = Get-CMManagementPoint -SiteSystemServerName "dist02.tsqa.contoso.com" -
SiteCode "CM1"
PS C:\> Remove-CMManagementPoint -InputObject $Mp
```

Related topics

[Add-CMManagementPoint](#)

[Get-CMManagementPoint](#)

[Get-CMManagementPointComponent](#)

Remove-CMOperatingSystemImage

Remove-CMOperatingSystemImage

Removes operating system images.

Syntax

Parameter Set: SearchByIdMandatory

```
Remove-CMOperatingSystemImage -Id <String[]> [-Force] [-SecuredScopeNames <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByNameMandatory

```
Remove-CMOperatingSystemImage -Name <String> [-Force] [-SecuredScopeNames <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Remove-CMOperatingSystemImage -InputObject <IResultObject> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Remove-CMOperatingSystemImage** cmdlet removes one or more operating system images from a Microsoft System Center 2012 Configuration Manager site. Operating system images are .wim format files and represent a compressed collection of reference files and folders that System Center 2012 Configuration Manager requires to successfully install and configure an operating system on a computer.

After you remove an operating system image, you cannot distribute the operating system image to distribution points.

Parameters

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Id<String[]>

Specifies an array of IDs of operating system images.

Aliases	PackageId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a **CMOperatingSystemImage** object. To obtain a **CMOperatingSystemImage** object, use the **Get-CMOperatingSystemImage** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies the name of an operating system image.

Aliases	none
Required?	true

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SecuredScopeNames<String>

Specifies names of security scopes. A security scope name can be Default or the name of a custom-created security scope.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Remove an operating system image

This command removes the operating system image named STANDARD_WIN7.

```
PS C:\> Remove-CMOperatingSystemImage -Name "STANDARD_WIN7"
```

Related topics

[Set-CMOperatingSystemImage](#)

[New-CMOperatingSystemImage](#)

[Get-CMOperatingSystemImageUpdateSchedule](#)

[Get-CMOperatingSystemImage](#)

Remove-CMOperatingSystemInstaller

Remove-CMOperatingSystemInstaller

Removes operating system installers.

Syntax

Parameter Set: SearchByIdMandatory

```
Remove-CMOperatingSystemInstaller -Id <String[]> [-Force] [-SecuredScopeNames <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByNameMandatory

```
Remove-CMOperatingSystemInstaller -Name <String> [-Force] [-SecuredScopeNames <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Remove-CMOperatingSystemInstaller -InputObject <IResultObject> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Remove-CMOperatingSystemInstaller** cmdlet removes one or more operating system installers from a Microsoft System Center 2012 Configuration Manager site. An operating system installer is an installation package that contains all the files that System Center 2012 Configuration Manager needs to install a Windows operating system on a reference computer.

After you remove an operating system installer, you cannot distribute the installation source files that are associated with the operating system installer to distribution points.

Parameters

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Id<String[]>

Specifies an array of IDs of operating system installers.

Aliases	PackageId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a **CMOperatingSystemInstaller** object. To obtain a **CMOperatingSystemInstaller** object, use the **Get-CMOperatingSystemInstaller** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies the name of an operating system installer.

Aliases	none
Required?	true
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SecuredScopeNames<String>

Specifies names of security scopes. A security scope name can be Default or the name of a custom-created security scope.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Remove an operating system installer

This command removes the operating system installer named INSTALL01.

```
PS C:\> Remove-CMOperatingSystemInstaller -Name "INSTALL01"
```

Related topics

[Get-CMOperatingSystemInstaller](#)

[New-CMOperatingSystemInstaller](#)

[Set-CMOperatingSystemInstaller](#)

Remove-CMOutOfBandServicePoint

Remove-CMOutOfBandServicePoint

Removes an out-of-band service point.

Syntax

Parameter Set: SearchByNameMandatory

```
Remove-CMOutOfBandServicePoint -SiteCode <String> -SiteSystemServerName <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Remove-CMOutOfBandServicePoint -InputObject <IResultObject> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Remove-CMOutOfBandServicePoint** cmdlet removes an out-of-band service point from Microsoft System Center 2012 Configuration Manager. An out-of-band service point is a site system role that provisions and configures Intel Active Management Technology (AMT)-based computers for System Center 2012 Configuration Manager.

After you remove an out-of-band service point, administrative users cannot provision and configure the Intel AMT-based computers that are associated with the out-of-band service point.

Parameters

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a **CMOutOfBandServicePoint** object. To obtain a **CMOutOfBandServicePoint** object, use the **Get-CMOutOfBandServicePoint** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteCode<String>

Specifies the site code of the Configuration Manager site that hosts the site system role.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteSystemServerName<String>

Specifies a fully qualified domain name (FQDN) of the server that hosts the site system role.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

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

Example 1: Remove an out-of-band service point

This command removes the out-of-band service point from the Configuration Manager site that has the site code CM1 on the site system server named cmcen-dist02.tsqa.contoso.com.

```
PS C:\> Remove-CMOutOfBandServicePoint -SiteSystemServerName "cmcen-dist02.tsqa.contoso.com"
-SiteCode "CM1"
```

Example 2: Remove an out-of-band service point by using an object variable

The first command gets the out-of-band service point from the Configuration Manager site that has the site code CM1 on the site system server named cmcen-dist02.tsqa.contoso.com. The command stores the results in the \$Osp variable.

The second command removes the out-of-band service point stored in the \$Osp variable.

```
PS C:\> $Osp = Get-CMOutOfBandServicePoint -SiteSystemServerName "cmcen-  
dist02.tsqa.contoso.com" -SiteCode "CM1"  
PS C:\> Remove-CMOutOfBandServicePoint -InputObject $Osp
```

Related topics

[Get-CMOutOfBandServicePoint](#)

[Set-CMOutOfBandServicePoint](#)

[Add-CMOutOfBandServicePoint](#)

Remove-CMPackage

Remove-CMPackage

Removes a Configuration Manager package.

Syntax

Parameter Set: SearchByIdMandatory

```
Remove-CMPackage -Id <String[]> [-Force] [-SecuredScopeNames <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByNameMandatory

```
Remove-CMPackage -Name <String> [-Force] [-SecuredScopeNames <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValue

```
Remove-CMPackage -InputObject <IResultObject> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Remove-CMPackage** cmdlet removes a package in Microsoft System Center 2012 Configuration Manager. You can delete a package from the site where it was created. System Center 2012 Configuration Manager cannot delete a package from a distribution point if a user has locked a network file.

When you remove a package, System Center 2012 Configuration Manager removes it from the database. If the package was sent to child sites, System Center 2012 Configuration Manager removes the package information at those child sites. If a compressed version of source files for the package exists, System Center 2012 Configuration Manager deletes the compressed file from the site server.

Parameters

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Id<String[]>

Specifies an array of package IDs.

Aliases	Packageld
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a **CMPackage** object. To obtain a **CMPackage** object, use the **Get-CMPackage** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies an array of package names.

Aliases	none
Required?	true

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SecuredScopeNames<String>

Specifies names of security scopes. A security scope name can be Default or the name of a custom-created security scope.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Remove a package

This command removes the package that has the ID CM10000D.

```
PS C:\> Remove-CMPackage -Id "CM10000D"
```

Example 2: Remove a package by using an object variable

The first command gets the package that has the ID CM10000D and assigns the results to the \$Pkg variable.

The second command removes the package stored in the \$Pkg variable.

```
PS C:\> $Pkg = Get-CMPackage -Id "CM10000D"  
PS C:\> Remove-CMPackage -InputObject $Pkg
```

Related topics

[Export-CMPackage](#)

[Get-CMPackage](#)

[Import-CMPackage](#)

[New-CMPackage](#)

[Set-CMPackage](#)

Remove-CMProgram

Remove-CMProgram

Removes programs from a Configuration Manager package.

Syntax

Parameter Set: SearchByIdAndNameMandatory

```
Remove-CMProgram -PackageId <String> -ProgramName <String> [-Force] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Remove-CMProgram -InputObject <IResultObject> [-Force] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Detailed Description

The **Remove-CMProgram** cmdlet removes one or more programs from a Microsoft System Center 2012 Configuration Manager package. Programs are commands that are associated with a System Center 2012 Configuration Manager package. Programs identify the actions that occur when the client receives the client package. You can associate multiple programs with the same package.

When you remove a program from a package, System Center 2012 Configuration Manager updates the package information in the System Center 2012 Configuration Manager site database.

System Center 2012 Configuration Manager removes all of the advertisements for this program from the database and removes the advertisements from clients that have received them. If System Center 2012 Configuration Manager has already run the advertised program on the client computer, System Center 2012 Configuration Manager does not remove the software.

Parameters

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a **CMProgram** object. To obtain a **CMProgram** object, use the **Get-CMProgram** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PackageId<String>

Specifies the package that contains the program by using an ID.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ProgramName<String>

Specifies the program within the package by using a name.

Aliases	none
Required?	true
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Remove a program by using a name and an ID

This command removes the program named ProgramD02 from the package that has the ID ST10000F.

```
PS C:\> Remove-CMProgram -PackageId "ST10000F" -ProgramName "ProgramD02"
```

Example 2: Remove a program by using an object variable

The first command gets the program named ProgramD02 in the package that has the ID ST10000F and assigns the results to the \$Prog variable.

The second command removes program stored in \$Prog.

```
PS C:\> $Prog = Get-CMProgram -Name "ProgramD02" -PackageId "ST10000F"  
PS C:\> Remove-CMProgram -InputObject $Prog
```

Related topics

[Disable-CMProgram](#)

[Enable-CMProgram](#)

[Get-CMProgram](#)

[New-CMProgram](#)

[Set-CMProgram](#)

Remove-CMRemoteConnectionProfileConfigurationItem

Remove-CMRemoteConnectionProfileConfigurationItem

Removes a remote connection profile.

Syntax

Parameter Set: SearchByNameMandatory

```
Remove-CMRemoteConnectionProfileConfigurationItem -Name <String[]> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Remove-CMRemoteConnectionProfileConfigurationItem -Id <String[]> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Remove-CMRemoteConnectionProfileConfigurationItem -InputObject <IResultObject> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Remove-CMRemoteConnectionProfileConfigurationItem** cmdlet removes a remote connection profile. Clients download the updated profile on the regular schedule.

Parameters

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-Id<String[]>

Specifies an array of IDs for remote connection profiles.

Aliases	CIId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a remote connection profile object. To obtain a remote connection profile, use the **Get-CMRemoteConnectionProfileConfigurationItem** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String[]>

Specifies an array of names of remote connection profiles.

Aliases	LocalizedDisplayName
Required?	true
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
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

[Copy-CMRemoteConnectionProfileConfigurationItem](#)

[Get-CMRemoteConnectionProfileConfigurationItem](#)

[New-CMRemoteConnectionProfileConfigurationItem](#)

[Set-CMRemoteConnectionProfileConfigurationItem](#)

Remove-CMReportingServicePoint

Remove-CMReportingServicePoint

Removes a reporting service point.

Syntax

Parameter Set: SearchByNameMandatory

```
Remove-CMReportingServicePoint -SiteCode <String> -SiteSystemServerName <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Remove-CMReportingServicePoint -InputObject <IResultObject> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Remove-CMReportingServicePoint** cmdlet removes a reporting service point from a Microsoft System Center 2012 Configuration Manager site. The reporting service point is a site system role that is installed on a server that is running Microsoft SQL Server Reporting Services.

After you remove a reporting service point from a System Center 2012 Configuration Manager site, you cannot manage reports in System Center 2012 Configuration Manager at the site.

Parameters

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a **CMReportingServicePoint** object. To obtain a **CMReportingServicePoint** object, use the **Get-CMReportingServicePoint** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteCode<String>

Specifies the site code of the Configuration Manager site that hosts the site system role.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteSystemServerName<String>

Specifies a fully qualified domain name (FQDN) of the server that hosts the site system role.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Remove a reporting service point

This command removes the reporting service point from the System Center 2012 Configuration Manager site that has the site code CM1 on the site system server named CMCEN-DIST02.TSQA.CORP.CONTOSCO.COM.

```
PS C:\> Remove-CMReportingServicePoint -SiteCode "CM1" -SiteSystemServerName "CMCEN-DIST02.TSQA.CORP.CONTOSCO.COM"
```

Example 2: Remove a reporting service point by using an object variable

The first command gets the reporting service point from the System Center 2012 Configuration Manager site that has the site code CM1 on the site system server named CMCEN-DIST02.TSQA.CORP.CONTOSCO.COM. The command stores the results in the \$Rsp variable.

The second command removes the reporting service point stored in the \$Rsp variable.

```
PS C:\> $Rsp = Get-CMReportingServicePoint -SiteCode "CM1" -SiteSystemServerName "CMCEN-DIST02.TSQA.CORP.CONTOSCO.COM"
```

```
PS C:\> Remove-CMReportingServicePoint -InputObject $Rsp
```

Related topics

[Add-CMReportingServicePoint](#)

[Get-CMReportingServicePoint](#)

[Set-CMReportingServicePoint](#)

Remove-CMSecondarySite

Remove-CMSecondarySite

Removes a secondary site from Configuration Manager.

Syntax

Parameter Set: SearchBySiteCodeMandatory

```
Remove-CMSecondarySite -Action {Delete | Uninstall} -SiteCode <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByNameMandatory

```
Remove-CMSecondarySite -Action {Delete | Uninstall} -Name <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Remove-CMSecondarySite -Action {Delete | Uninstall} -InputObject <IResultObject> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Remove-CMSecondarySite** cmdlet removes a secondary site from Microsoft System Center 2012 Configuration Manager. A secondary site has no site database of its own. Instead it is connected to a primary site and sends client data to the primary site for storage.

Parameters

-Action<ActionType>

Specifies an action type for the deletion. Valid values are:

-- Delete. Removes the reference to the secondary site from the database.

-- Uninstall. Removes the reference to the secondary site from the database and triggers an uninstall action at the secondary site server.

The acceptable values for this parameter are:

Delete	
Uninstall	

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a secondary site object. To obtain this object, use the [New-CMSecondarySite](#) cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies the name of a secondary site.

Aliases	SiteName
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteCode<String>

Specifies a code for a secondary site.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Remove a secondary site upgrade by using a site name

This command deletes a secondary site named ClientSecSiteUpgrade03. Because the *Force* parameter is not specified, you must confirm the action before it is performed.

```
PS C:\> Remove-CMSecondarySite -Action Delete -SiteName "ClientSecSiteUpgrade03"
```

Related topics

[Invoke-CMSecondarySiteUpgrade](#)

[New-CMSecondarySite](#)

Remove-CMSecurityRole

Remove-CMSecurityRole

Removes custom security roles from Configuration Manager.

Syntax

Parameter Set: SearchByIdMandatory

```
Remove-CMSecurityRole -Id <String[]> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByNameMandatory

```
Remove-CMSecurityRole -Name <String[]> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Remove-CMSecurityRole -InputObject <IResultObject> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Remove-CMSecurityRole** cmdlet removes custom security roles from Microsoft System Center 2012 Configuration Manager. Specify the name or ID of a security role you want to remove or use the [Get-CMSecurityRole](#) cmdlet to obtain one.

System Center 2012 Configuration Manager uses security roles, along with security scopes and collections, to define an administrative scope for each administrative user. System Center 2012 Configuration Manager provides several built-in security roles. To create a custom security role, copy an existing security role, and then modifying the copy. You can copy a security role by using the [Copy-CMSecurityRole](#) cmdlet.

You can use the **Remove-CMSecurityRole** cmdlet to remove old, unneeded custom security roles. You cannot remove built-in security roles. Every administrative user must have at least one security role. Before you remove a security role, make sure every user has a role in addition to the one you remove.

Parameters

-Force

Performs the action without a confirmation message.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Id<String[]>

Specifies an array of IDs of security roles.

Aliases	RoleId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a security role object. To obtain a security role object, use the [Get-CMSecurityRole](#) cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String[]>

Specifies an array of names of security roles.

Aliases	RoleName
---------	----------

Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	true

-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

Example 1: Remove a security role by using a name

This command removes a security role named MainSecurityRole from System Center 2012 Configuration Manager. The command uses the *Force* parameter, so it does not prompt you for confirmation.

```
PS C:\> Remove-CMSecurityRole -Name "MainSecurityRole" -Force
```

Example 2: Remove security roles by using a variable

The first command uses the **Get-CMSecurity** cmdlet to get each security role that has a name that ends in Role. It stores them in the \$Roles variable.

The second command removes each security role stored in the \$Roles variable.

```
PS C:\> $Roles = Get-CMSecurityRole -Name *Role
PS C:\> Remove-CMSecurityRole -SecurityRole $Roles
```

Related topics

[Copy-CMSecurityRole](#)

[Export-CMSecurityRole](#)

[Import-CMSecurityRole](#)

[Get-CMSecurityRole](#)

[Set-CMSecurityRole](#)

Remove- CMSecurityRoleFromAdministrativeUser

Remove-CMSecurityRoleFromAdministrativeUser

Removes the association between a security role and an administrative user.

Syntax

Parameter Set: RemoveRoleFromAdminByName_Name

```
Remove-CMSecurityRoleFromAdministrativeUser -AdministrativeUserName <String> -RoleName  
<String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: RemoveRoleFromAdminById_Id

```
Remove-CMSecurityRoleFromAdministrativeUser -AdministrativeUserId <Int32> -RoleId <String>  
[-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: RemoveRoleFromAdminById_Name

```
Remove-CMSecurityRoleFromAdministrativeUser -AdministrativeUserName <String> -RoleId  
<String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: RemoveRoleFromAdminById_Object

```
Remove-CMSecurityRoleFromAdministrativeUser -AdministrativeUser <IResultObject> -RoleId  
<String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: RemoveRoleFromAdminByName_Id

```
Remove-CMSecurityRoleFromAdministrativeUser -AdministrativeUserId <Int32> -RoleName <String>  
[-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: RemoveRoleFromAdminByName_Object

```
Remove-CMSecurityRoleFromAdministrativeUser -AdministrativeUser <IResultObject> -RoleName  
<String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: RemoveRoleFromAdminByObject_Id

```
Remove-CMSecurityRoleFromAdministrativeUser -AdministrativeUserId <Int32> -Role  
<IResultObject> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: RemoveRoleFromAdminByObject_Name

```
Remove-CMSecurityRoleFromAdministrativeUser -AdministrativeUserName <String> -Role  
<IResultObject> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: RemoveRoleFromAdminByObject_Object

```
Remove-CMSecurityRoleFromAdministrativeUser -AdministrativeUser <IResultObject> -Role  
<IResultObject> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Remove-CMSecurityRoleFromAdministrativeUser** cmdlet removes the association between one or more security roles and an administrative user. After you remove the association of a security role with an administrative user, the administrative user cannot view the objects in Microsoft System Center 2012 Configuration Manager that are associated with the security role, and no longer has the permission to perform the tasks that are related to those objects.

Parameters

-AdministrativeUser<IResultObject>

Specifies a **CMAdministrativeUser** object. To obtain a **CMAdministrativeUser** object, use the [Get-CMAdministrativeUser](#) cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AdministrativeUserId<Int32>

Specifies the ID of an administrative user.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AdministrativeUserName<String>

Specifies the name of an administrative user.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Role<IResultObject>

Specifies a **CMSecurityRole** object. To obtain a **CMSecurityRole** object, use the [Get-CMSecurityRole](#) cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RoleId<String>

Specifies the ID of a role.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RoleName<String>

Specifies the name of a role.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Remove a security role from an administrative user

This command removes the association between the security role named Security Update Manager and the administrative user named Team04\TeamAdmin.

```
PS C:\> Remove-CMSecurityRoleFromAdministrativeUser -AdministrativeUserName  
"Team04\TeamAdmin" -RoleName "Security Update Manager"
```

Related topics

[Add-CMSecurityRoleToAdministrativeUser](#)

[Get-CMSecurityRole](#)

[Set-CMSecurityRole](#)

[Copy-CMSecurityRole](#)

[Remove-CMSecurityRole](#)

[Import-CMSecurityRole](#)

[Export-CMSecurityRole](#)

Remove-CMSecurityScope

Remove-CMSecurityScope

Removes a security scope.

Syntax

Parameter Set: SearchByIdMandatory

```
Remove-CMSecurityScope -Id <String[]> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByNameMandatory

```
Remove-CMSecurityScope -Name <String[]> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Remove-CMSecurityScope -InputObject <IResultObject> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Remove-CMSecurityScope** cmdlet removes a security scope in Microsoft System Center 2012 Configuration Manager. Before you can remove a security scope, you must remove the association between administrative users and the security scope. You can use the **Remove-CMSecurityScopeFromAdministrativeUser** cmdlet to remove the association between a security scope and an administrative user.

Parameters

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Id<String[]>

Specifies an array of IDs of security scopes.

Aliases	CategoryId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a **CMSecurityScope** object. To obtain a **CMSecurityScope** object, use the **Get-
CMSecurityScope** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String[]>

Specifies an array of names of security scopes.

Aliases	CategoryName
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	true
-----------------------------	------

-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

Example 1: Remove a security scope by using a name

This command removes the security scope named ScopeT02.

```
PS C:\> Remove-CMSecurityScope -Name "ScopeT02"
```

Example 2: Remove a security scope by using an ID

This command removes the security scope that has the ID CM100004.

```
PS C:\> Remove-CMSecurityScope -Id "CM100004"
```

Example 3: Remove a security scope by using an object variable

The first command gets the security scopes that have a name that starts with NewS and assigns the results to the \$SecurityScope variable.

The second command removes the security scopes stored in the \$SecurityScope variable.

```
PS C:\> $SecurityScope = Get-CMSecurityScope -Name NewS*
```

```
PS C:\> Remove-CMSecurityScope -InputObject $SecurityScope
```

Related topics

[Get-CMSecurityScope](#)

[New-CMSecurityScope](#)

[Set-CMSecurityScope](#)

[Remove-CMSecurityScopeFromAdministrativeUser](#)

Remove- CMSecurityScopeFromAdministrativeUser

Remove-CMSecurityScopeFromAdministrativeUser

Removes the association between security scopes and an administrative user.

Syntax

Parameter Set: RemoveScopeFromAdminByName_Name

```
Remove-CMSecurityScopeFromAdministrativeUser -AdministrativeUserName <String> -  
SecurityScopeName <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: RemoveScopeFromAdminById_Id

```
Remove-CMSecurityScopeFromAdministrativeUser -AdministrativeUserId <Int32> -SecurityScopeId  
<String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: RemoveScopeFromAdminById_Name

```
Remove-CMSecurityScopeFromAdministrativeUser -AdministrativeUserName <String> -  
SecurityScopeId <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: RemoveScopeFromAdminById_Object

```
Remove-CMSecurityScopeFromAdministrativeUser -AdministrativeUser <IResultObject> -  
SecurityScopeId <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: RemoveScopeFromAdminByName_Id

```
Remove-CMSecurityScopeFromAdministrativeUser -AdministrativeUserId <Int32> -  
SecurityScopeName <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: RemoveScopeFromAdminByName_Object

```
Remove-CMSecurityScopeFromAdministrativeUser -AdministrativeUser <IResultObject> -  
SecurityScopeName <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: RemoveScopeFromAdminByObject_Id

```
Remove-CMSecurityScopeFromAdministrativeUser -AdministrativeUserId <Int32> -SecurityScope  
<IResultObject> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: RemoveScopeFromAdminByObject_Name

```
Remove-CMSecurityScopeFromAdministrativeUser -AdministrativeUserName <String> -SecurityScope  
<IResultObject> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: RemoveScopeFromAdminByObject_Object

```
Remove-CMSecurityScopeFromAdministrativeUser -AdministrativeUser <IResultObject> -  
SecurityScope <IResultObject> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Remove-CMSecurityScopeFromAdministrativeUser** cmdlet removes the association between one or more security scopes and an administrative user.

After you remove the association between a security scope and an administrative user, the administrative user cannot view the objects in Microsoft System Center 2012 Configuration Manager that are associated with the security scope, and no longer has the permission to perform the tasks that are related to those objects.

Parameters

-AdministrativeUser<IResultObject>

Specifies a **CMAdministrativeUser** object. To obtain a **CMAdministrativeUser** object, use the [Get-CMAdministrativeUser](#) cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AdministrativeUserId<Int32>

Specifies the ID of an administrative user.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AdministrativeUserName<String>

Specifies the name of an administrative user.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SecurityScope<IResultObject>

Specifies a security scope object. To obtain a security scope object, use the **Get-CMSecurityScope** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SecurityScopeld<String>

Specifies the ID of a security scope.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SecurityScopeName<String>

Specifies the name of a security scope. A security scope name can be Default or the name of a custom security scope.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Remove a security scopes from an administrative user

This command removes the association between the security scope named SecScope02 and the administrative user named Team04\TeamAdmin.

```
PS C:\> Remove-CMSecurityScopeFromAdministrativeUser -AdministrativeUserName  
"Team04\TeamAdmin" -SecurityScopeName "SecScope02"
```

Related topics

[Add-CMSecurityScopeToAdministrativeUser](#)

[Get-CMSecurityScope](#)

[New-CMSecurityScope](#)

[Set-CMSecurityScope](#)

[Remove-CMSecurityScope](#)

Remove-CMSoftwareMeteringRule

Remove-CMSoftwareMeteringRule

Removes Configuration Manager software metering rules.

Syntax

Parameter Set: SearchByIdMandatory

```
Remove-CMSoftwareMeteringRule -Id <String[]> [-Force] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByNameMandatory

```
Remove-CMSoftwareMeteringRule -ProductName <String> [-Force] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Remove-CMSoftwareMeteringRule -InputObject <IResultObject> [-Force] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Detailed Description

The **Remove-CMSoftwareMeteringRule** cmdlet removes one or more software metering rules from Microsoft System Center 2012 Configuration Manager.

Software metering monitors and collects software usage data from System Center 2012 Configuration Manager clients, such as when clients began using a particular software program and how long users have worked with that software. You can create software metering rules that specify which software to monitor.

You can specify rules to disable by ID or by product name, or use the **Get-CMSoftwareMeteringRule** cmdlet. You can use the **Disable-CMSoftwareMeteringRule** to temporarily suspend a rule.

For more information about software metering in System Center 2012 Configuration Manager, see [Introduction to Software Metering in Configuration Manager](http://go.microsoft.com/fwlink/?LinkId=268432) (<http://go.microsoft.com/fwlink/?LinkId=268432>) on TechNet.

Parameters

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Id<String[]>

Specifies an array of IDs for software metering rules.

Aliases	RuleId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a software metering rule object. To obtain a software metering rule object, use the **Get-SoftwareMeteringRule** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ProductName<String>

Specifies a name for a product that a rule meters.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Remove rules for a product

This command removes any software metering rules for a product called Accounting Package. In this example, there are two rules for that product. The command does not include the *Force* parameter, so the cmdlet prompts for confirmation for both rules.

```
PS C:\> Remove-CMSoftwareMeteringRule -ProductName "Accounting Package"
```

```
Remove
```

```
Are you sure you wish to remove SoftwareMeteringRule: RuleID=16777220?
```

```
[Y] Yes [N] No [S] Suspend [?] Help (default is "Y"): Y
```

```
Remove
```

```
Are you sure you wish to remove SoftwareMeteringRule: RuleID=16777221?
```

```
[Y] Yes [N] No [S] Suspend [?] Help (default is "Y"): Y
```

Related topics

[Disable-CMSoftwareMeteringRule](#)

[Enable-CMSoftwareMeteringRule](#)

[Get-CMSoftwareMeteringRule](#)

[New-CMSoftwareMeteringRule](#)

[Set-CMSoftwareMeteringRule](#)

Remove-CMSoftwareUpdateAutoDeploymentRule

Remove-CMSoftwareUpdateAutoDeploymentRule

Removes Configuration Manager deployment rules for automatic software updates.

Syntax

Parameter Set: SearchByIdMandatory

```
Remove-CMSoftwareUpdateAutoDeploymentRule -Id <String[]> [-Force] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByNameMandatory

```
Remove-CMSoftwareUpdateAutoDeploymentRule -Name <String> [-Force] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Remove-CMSoftwareUpdateAutoDeploymentRule -InputObject <IResultObject> [-Force] [-Confirm] [  
-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Remove-CMSoftwareUpdateAutoDeploymentRule** cmdlet removes specified Microsoft System Center 2012 Configuration Manager deployment rules for automatic software updates.

System Center 2012 Configuration Manager uses rules to manage automatic deployment of software updates. When a rule runs, System Center 2012 Configuration Manager adds updates that qualify for the rule to a software update group. The System Center 2012 Configuration Manager server downloads content files and copies them to distribution points, and then updates client computers.

You can specify rules to remove by ID or by name, or specify a rule object by using the **Get-CMSoftwareUpdateAutoDeploymentRule** cmdlet. This cmdlet deletes rules permanently. You can use the **Disable-CMSoftwareUpdateAutoDeploymentRule** cmdlet to suspend a rule.

Parameters

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Id<String[]>

Specifies an array of IDs for rules for automatic deployment of software updates. This value is the **AutoDeploymentID** property of the deployment rule object.

Aliases	AutoDeploymentId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a software update automatic deployment rule object. To obtain a deployment rule object, use the **Get-CMSoftwareUpdateAutoDeploymentRule** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies a name of a rule for automatic deployment of software updates.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Remove a deployment rule by name

This command removes a rule named Weekly Driver Updates. Because the command does not include the *Force* parameter, the cmdlet prompts you before it deletes the rule.

```
PS C:\> Remove-CMSoftwareUpdateAutoDeploymentRule -Name "Weekly Driver Updates"
```

Remove

Are you sure you wish to remove SoftwareUpdateAutoDeploymentRule: Name="Weekly Driver Updates"?

[Y] Yes [N] No [S] Suspend [?] Help (default is "Y"):

Example 2: Remove a deployment rule by ID

This command disables a deployment rule that has the ID 16777217. This command includes the *Force* parameter, so the cmdlet does not prompt you before it removes the rule.

```
PS C:\> Remove-CMSoftwareUpdateAutoDeploymentRule -Id "16777217" -Force
```

Example 3: Remove a deployment rule by using a variable

The first command gets a deployment rule that has the specified name and stores it in the `$CMSUADR` variable.

The second command removes the rule stored in the variable.

```
PS C:\> $CMSUADR = Get-CMSoftwareUpdateAutoDeploymentRule -Name "Weekly Driver Updates"
```

```
PS C:\> Remove-CMSoftwareUpdateAutoDeploymentRule -InputObject $CMSUADR -Force
```

Related topics

[Disable-CMSoftwareUpdateAutoDeploymentRule](#)

[Enable-CMSoftwareUpdateAutoDeploymentRule](#)

[Get-CMSoftwareUpdateAutoDeploymentRule](#)

[Invoke-CMSoftwareUpdateAutoDeploymentRule](#)

Remove-CMSoftwareUpdateDeploymentPackage

Remove-CMSoftwareUpdateDeploymentPackage

Removes a deployment package.

Syntax

Parameter Set: SearchByIdMandatory

```
Remove-CMSoftwareUpdateDeploymentPackage -Id <String[]> [-Force] [-SecuredScopeNames <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByNameMandatory

```
Remove-CMSoftwareUpdateDeploymentPackage -Name <String> [-Force] [-SecuredScopeNames <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Remove-CMSoftwareUpdateDeploymentPackage -InputObject <IResultObject> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Remove-CMSoftwareUpdateDeploymentPackage** cmdlet removes a software update deployment package from the site server and all child sites. A

CMSoftwareUpdateDeploymentPackage object contains one or more software updates that Microsoft System Center 2012 Configuration Manager deploys to a collection of computers. Once the deployment package is removed, clients cannot install the software updates.

Parameters

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Id<String[]>

Specifies an array of IDs of deployment packages.

Aliases	PackageId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a **CMSoftwareUpdateDeploymentPackage** object. To obtain an **CMSoftwareUpdateDeploymentPackage** object, use the **Get-
CMSoftwareUpdateDeploymentPackage** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies a name of a deployment package.

Aliases	none
---------	------

Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SecuredScopeNames<String>

Specifies an array of security scopes for the deployment package. The default value is Default.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Remove a software package by using an ID

This command removes the software package that has the ID ST10000C.

```
PS C:\> Remove-CMSoftwareUpdateDeploymentPackage -PackageID "ST10000C"
```

Related topics

[Get-CMSoftwareUpdateDeploymentPackage](#)

[Set-CMSoftwareUpdateDeploymentPackage](#)

Remove-CMSoftwareUpdateGroup

Remove-CMSoftwareUpdateGroup

Removes Configuration Manager software update groups.

Syntax

Parameter Set: SearchByIdMandatory

```
Remove-CMSoftwareUpdateGroup -Id <String[]> [-Force] [-SecuredScopeNames <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByNameMandatory

```
Remove-CMSoftwareUpdateGroup -Name <String[]> [-Force] [-SecuredScopeNames <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Remove-CMSoftwareUpdateGroup -InputObject <IResultObject> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Remove-CMSoftwareUpdateGroup** cmdlet removes software update groups from Microsoft System Center 2012 Configuration Manager. You can specify each software update group that you are removing by using the group IDs or names. Or, when you remove a software update group, you can use the **Get-CMSoftwareUpdateGroup** cmdlet to return a software update group object and use that object to specify the group that you want to remove.

Parameters

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Id<String[]>

Specifies an array of software update group IDs.

Aliases	CId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies the software update group object to remove. To obtain a software update group object, use the **Get-CMSoftwareUpdateGroup** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String[]>

Specifies an array of software update group names.

Aliases	LocalizedDisplayName
Required?	true
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SecuredScopeNames<String>

Specifies a security scope name. This name identifies the security scope from which you remove the software update groups. A security scope name can be either Default or the name of a custom security scope.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Remove a software update group by using an ID

This command removes the software update group that has the ID ST10000B.

```
PS C:\> Remove-CMSoftwareUpdateGroup -Id "ST10000B"
```

Example 2: Remove a software update group by using a name

This command removes the software update group named SUGroupD01.

```
PS C:\> Remove-CMSoftwareUpdateGroup -Name "SUGroupD01"
```

Example 3: Remove a software update group by using an object variable

The first command gets the software update group that has the ID ST10000B and stores it in the variable \$SubObj.

The second command removes the software update group by using the \$SubObj variable.

```
PS C:\> $SubObj=Get-CMSoftwareUpdateGroup -Id "ST10000B"  
PS C:\> Remove-CMSoftwareUpdateGroup -SoftwareUpdateGroup $SubObj
```

Related topics

[Get-CMSoftwareUpdateGroup](#)

[New-CMSoftwareUpdateGroup](#)

[Set-CMSoftwareUpdateGroup](#)

Remove-CMSoftwareUpdatePoint

Remove-CMSoftwareUpdatePoint

Removes a software update point site system role from Configuration Manager.

Syntax

Parameter Set: SearchByNameMandatory

```
Remove-CMSoftwareUpdatePoint -SiteCode <String> -SiteSystemServerName <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Remove-CMSoftwareUpdatePoint -InputObject <IResultObject> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Remove-CMSoftwareUpdatePoint** cmdlet removes a software update point site system role from Microsoft System Center 2012 Configuration Manager.

A software update point is a site server role that hosts software updates. System Center 2012 Configuration Manager clients connect to a software update point to get available updates. The software update point interacts with Windows Server Update Services (WSUS) to configure update settings, request synchronization to the update source, and to synchronize software updates from the WSUS database.

You can specify a software update point to remove by site code and the name of the computer that hosts the site system role. You can also use the **Get-CMSoftwareUpdatePoint** cmdlet to obtain a software update point.

Parameters

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a software update point object. To obtain a software update point object, use the **Get-CMSoftwareUpdatePoint** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteCode<String>

Specifies a site code for a Configuration Manager site.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteSystemServerName<String>

Specifies the name of a computer that hosts the software update point site system role.

Aliases	none
Required?	true

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Remove a software update point

The command removes a software update point. The cmdlet requires both the site code and the name. Because the command does not include the *Force* parameter, the cmdlet prompts you for confirmation.

```
PS C:\> Remove-CMSoftwareUpdatePoint -SiteCode "CM1" -SiteSystemServerName  
"UpdateSystem.Western.Contoso.com"
```

Example 2: Remove a software update point by using a variable

The first command gets a software update point and saves it to the \$CMSUP variable.

The second command removes the software update point saved in the \$CMSUP variable. This command uses the *Force* parameter, so the cmdlet does not prompt you for confirmation.

```
PS C:\> $CMSUP = Get-CMSoftwareUpdatePoint -SiteCode "CM1" -SiteSystemServerName  
"UpdateSystem.Western.Contoso.com"  
PS C:\> Remove-CMSoftwareUpdatePoint -InputObject $CMSUP -Force
```

Related topics

[Add-CMSoftwareUpdatePoint](#)

[Get-CMSoftwareUpdatePoint](#)

[Set-CMSoftwareUpdatePoint](#)

Remove-CMStateMigrationPoint

Remove-CMStateMigrationPoint

Removes a state migration point from a Configuration Manager site.

Syntax

Parameter Set: SearchByNameMandatory

```
Remove-CMStateMigrationPoint -SiteCode <String> -SiteSystemServerName <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Remove-CMStateMigrationPoint -InputObject <IResultObject> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Remove-CMStateMigrationPoint** cmdlet removes a state migration point from a Microsoft System Center 2012 Configuration Manager site. This site system role stores user information while you perform an operating system deployment. If you remove a state migration point, you also remove all associated stored user information.

Each state migration point can be a member of only one System Center 2012 Configuration Manager site.

Parameters

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a state migration point object. To obtain a state migration point object, use the **Get-CMStateMigrationPoint** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteCode<String>

Specifies a site code for a Configuration Manager site.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteSystemServerName<String>

Specifies the host name for a state migration point.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

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

Example 1: Remove a specified migration point

This command removes a state migration point that belongs to the site that has the site code CM1. The command specifies the name of computer that hosts the site system role.

```
PS C:\> Remove-CMStateMigrationPoint -SiteCode "CM1" -SiteSystemServerName  
"SMP01.Western.Contoso.com"
```

Example 2: Remove a migration point using a variable

The first command uses the **Get-CMStateMigrationPoint** to get a state migration point that belongs to the specified site and has the specified host name, and then stores that object in the \$CMSMP variable.

The second command removes the state migration point stored in the \$CMSMP variable.

```
PS C:\> $CMSMP = Get-CMStateMigrationPoint -SiteCode "CM1" -SiteSystemServerName  
"SMP01.Western.Contoso.com"
```

```
PS C:\> Remove-CMStateMigrationPoint -InputObject $CMSMP
```

Related topics

[Add-CMStateMigrationPoint](#)

[Get-CMStateMigrationPoint](#)

Remove-CMStatusFilterRule

Remove-CMStatusFilterRule

Removes a specified Configuration Manager filter rule for status messages.

Syntax

Parameter Set: SearchBySiteCodeMandatory

```
Remove-CMStatusFilterRule -Name <String> -SiteCode <String> [-Force] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByValue

```
Remove-CMStatusFilterRule -InputObject <IResultObject> [-Force] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Detailed Description

The **Remove-CMStatusFilterRule** cmdlet removes a specified Microsoft System Center 2012 Configuration Manager filter rule for status messages.

Status filter rules specify how System Center 2012 Configuration Manager responds to status messages. Each filter rule contains criteria and actions for status messages. You configure status filter rules for each site, not across all sites.

Use the rule name and site code to specify a rule to remove. This cmdlet deletes rules permanently. You can use the **Disable-CMStatusFilterRule** cmdlet to suspend a rule.

Parameters

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-InputObject<IResultObject>

Specifies a status filter rule object to remove. To obtain a status filter rule object, use the **Get-
CMStatusFilterRule** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies a name of a rule.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteCode<String>

Specifies a site code for the Configuration Manager site.

Aliases	none
Required?	true
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Remove a rule

This command removes a status filter rule that has the specified name in a site that has the site code CM1. The command includes the *Force* parameter, so the cmdlet does not prompt you for confirmation.

```
PS C:\> Remove-CMStatusFilterRule -Name "Status change to critical" -SiteCode "CM1" -Force
```

Related topics

[Disable-CMStatusFilterRule](#)

[Enable-CMStatusFilterRule](#)

[Get-CMStatusFilterRule](#)

[New-CMStatusFilterRule](#)

[Set-CMStatusFilterRule](#)

Remove-CMStatusMessageQuery

Remove-CMStatusMessageQuery

Removes a Configuration Manager status message query.

Syntax

Parameter Set: SearchByIdMandatory

```
Remove-CMStatusMessageQuery -Id <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByNameMandatory

```
Remove-CMStatusMessageQuery -Name <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Remove-CMStatusMessageQuery -InputObject <IResultObject> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Remove-CMStatusMessageQuery** cmdlet removes a status message query from Microsoft System Center 2012 Configuration Manager. Status message queries return status messages from the System Center 2012 Configuration Manager site database.

Parameters

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Id<String>

Specifies an ID of a status message query.

Aliases	QueryId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a status message query object. To obtain a status message query object, use the **Get-CMStatusMessageQuery** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies a name of a status message query.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Remove a named query

This command removes a query named All Audit Status Messages from a Specific Site. The command does not include the *Force* parameter, so the cmdlet prompts you before it removes the query.

```
PS C:\> Remove-CMStatusMessageQuery -Name "All Audit Status Messages from a Specific Site"
```

```
Remove
```

```
Are you sure you wish to remove StatusMessageQuery: Name="All Audit Status Messages from a Specific Site"?
```

```
[Y] Yes [N] No [S] Suspend [?] Help (default is "Y"):
```

Example 2: Remove a query that has a specified ID

This command removes the query that has an ID of CM100008. The command includes the *Force* parameter, so the cmdlet does not prompt you for confirmation.

```
PS C:\> Remove-CMStatusMessageQuery -Id "CM100008" -Force
```

Related topics

[Get-CMStatusMessageQuery](#)

[New-CMStatusMessageQuery](#)

[Set-CMStatusMessageQuery](#)

Remove-CMSystemHealthValidatorPoint

Remove-CMSystemHealthValidatorPoint

Removes a system health validator point from Configuration Manager.

Syntax

Parameter Set: SearchByNameMandatory

```
Remove-CMSystemHealthValidatorPoint -SiteCode <String> -SiteSystemServerName <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Remove-CMSystemHealthValidatorPoint -InputObject <IResultObject> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Remove-CMSystemHealthValidatorPoint** cmdlet removes a system health validator point from a Microsoft System Center 2012 Configuration Manager site. This site system role validates statements of health from a server that is running Network Policy Server (NPS). You can specify a validator point by site system name or site code or both or you can use the **Get-CMSystemHealthValidatorPoint** cmdlet.

Before you remove a system health validator point, make sure that there is another system health validator point for the site, or that the server that is running NPS has policies that grant network access and do not reference the System Center 2012 Configuration Manager.

Parameters

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-InputObject<IResultObject>

Specifies a system health validator point object. To obtain a system health validator point object, use the **Get-CMSystemHealthValidatorPoint** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteCode<String>

Specifies a site code for a Configuration Manager site.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteSystemServerName<String>

Specifies the host name for a system health validator point.

Aliases	none
Required?	true
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Remove a validator point

This command removes a system health validator point. The command specifies the site code and the name of the server that hosts that system role.

```
PS C:\> Remove-CMSystemHealthValidatorPoint -SiteCode "CM1" -SiteSystemServerName
"Test01.Western.Contoso.com"
```

Example 2: Remove a validator point by using a variable

The first command gets the system role that has the specified site code and host name and stores it in the \$CMSHVP variable.

The second command removes the system health validator point stored in the \$CMSHVP variable.

```
PS C:\> $CMSHVP = Get-CMSystemHealthValidatorPoint -SiteCode "CM1" -SiteSystemServerName  
"Test01.Western.Contoso.com"
```

```
PS C:\> Remove-CMSystemHealthValidatorPoint -InputObject $CMSHVP
```

Related topics

[Add-CMSystemHealthValidatorPoint](#)

[Get-CMSystemHealthValidatorPoint](#)

Remove-CMTaskSequence

Remove-CMTaskSequence

Removes a Configuration Manager task sequence.

Syntax

Parameter Set: SearchByIdMandatory

```
Remove-CMTaskSequence -TaskSequencePackageId <String[]> [-Force] [-SecuredScopeNames <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByNameMandatory

```
Remove-CMTaskSequence -Name <String> [-Force] [-SecuredScopeNames <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Remove-CMTaskSequence -InputObject <IResultObject> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Remove-CMTaskSequence** cmdlet removes a Microsoft System Center 2012 Configuration Manager task sequence. A task sequence includes configuration and operating system deployment settings for a System Center 2012 Configuration Manager client computer.

You can specify a name or ID to remove a specific sequence or use the **Get-CMTaskSequence** cmdlet to obtain a sequence. To remove a sequence only if it has a particular security scope, you can specify a security scope along with a name or ID.

This cmdlet removes a sequence permanently. Use the **Disable-CMTaskSequence** cmdlet to stop advertising a task sequence to clients temporarily.

Parameters

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a task sequence object. To obtain a task sequence object, use the **Get-CMTaskSequence** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies a name for a task sequence.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SecuredScopeNames<String>

Specifies a name of a security scope.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-TaskSequencePackageId<String[]>

Specifies an array of IDs of task sequences.

Aliases	PackageId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Remove a task sequence by using a variable

The first command gets the task sequence object that has the specified name and stores it in the \$CMTS variable.

The second command removes the object stored in \$CMTS.

```
PS C:\> $CMTS = Get-CMTaskSequence -Name "General Sequence 11"
PS C:\> Remove-CMTaskSequence -InputObject $CMTS
```

Remove

```
Are you sure you wish to remove TaskSequence: Name="General Sequence 11"?
[Y] Yes [N] No [S] Suspend [?] Help (default is "Y"):
```

Example 2: Remove a specified task sequence

This command removes the task sequence object that has the specified ID. This command uses the *Force* parameter, so the cmdlet does not prompt you for confirmation.

```
PS C:\> Remove-CMTaskSequence -Id "CM10000A" -Force
```

Related topics

[Disable-CMTaskSequence](#)

[Enable-CMTaskSequence](#)

[Export-CMTaskSequence](#)

[Get-CMTaskSequence](#)

[Import-CMTaskSequence](#)

[New-CMTaskSequence](#)

[Set-CMTaskSequence](#)

Remove- CMTrustedRootCertificateProfileConfigurationItem

Remove-CMTrustedRootCertificateProfileConfigurationItem

Removes a root certificate profile.

Syntax

Parameter Set: SearchByNameMandatory

```
Remove-CMTrustedRootCertificateProfileConfigurationItem -Name <String[]> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Remove-CMTrustedRootCertificateProfileConfigurationItem -Id <String[]> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Remove-CMTrustedRootCertificateProfileConfigurationItem -InputObject <IResultObject> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Remove-CMTrustedRootCertificateProfileConfigurationItem** cmdlet removes a root certificate profile.

Parameters

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Id<String[]>

Specifies an array of IDs of root certificate profiles.

Aliases	CId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a root certificate profile object. To obtain a root certificate profile object use the **Get-CMTrustedRootCertificateProfileConfigurationItem** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String[]>

Specifies an array of names of root certificate profiles.

Aliases	LocalizedDisplayName
Required?	true
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

[Copy-CMTrustedRootCertificateProfileConfigurationItem](#)

[Get-CMTrustedRootCertificateProfileConfigurationItem](#)

[New-CMTrustedRootCertificateProfileConfigurationItem](#)

[Set-CMTrustedRootCertificateProfileConfigurationItem](#)



Remove-CMUser

Remove-CMUser

Removes Configuration Manager user accounts.

Syntax

Parameter Set: SearchByIdMandatory

```
Remove-CMUser -Id <String[]> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByNameMandatory

```
Remove-CMUser -Name <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Remove-CMUser -InputObject <IResultObject> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Remove-CMUser** cmdlet removes Microsoft System Center 2012 Configuration Manager user accounts. You can specify user accounts to remove by ID or by name, or you can use the **Get-CMUser** cmdlet to obtain user accounts to remove.

Parameters

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Id<String[]>

Specifies an array of IDs for Configuration Manager user accounts.

Aliases	ResourceId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a Configuration Manager user account object. To obtain a Configuration Manager user account object, use the **Get-CMUser** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies the name of a Configuration Manager user account.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Remove a user account by using an ID

This command removes the user account that has the ID 2063597584.

```
PS C:\> Remove-CMUser -Id "2063597584"
```

Example 2: Remove a user account by name

This command removes a user account that has the specified name.

```
PS C:\> Remove-CMUser -Name "FAREAST\001D$ (001D$)"
```

Example 3: Remove user accounts by using a variable

This example removes user accounts that are part of the collection that has the ID SMS00004. The first command uses the **Get-CMUser** cmdlet to get the user accounts that belong to the collection with the specified ID, and stores those user accounts in the \$CMUsers variable.

The second command removes the user accounts stored in the \$CMUsers variable.

```
PS C:\> $CMUsers = Get-CMUser -CollectionID "SMS00004"
```

```
PS C:\> Remove-CMUser -InputObject $CMUsers
```

Related topics

[Get-CMUser](#)

Remove-CMUserAffinityFromDevice

Remove-CMUserAffinityFromDevice

Removes a primary user from one or more devices in the Configuration Manager hierarchy.

Syntax

Parameter Set: RemoveUserAffinityByDeviceName

```
Remove-CMUserAffinityFromDevice -DeviceName <String[]> [-Force] [-UserId <String> ] [-UserName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: RemoveUserAffinityByDeviceId

```
Remove-CMUserAffinityFromDevice -DeviceId <String[]> [-Force] [-UserId <String> ] [-UserName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Remove-CMUserAffinityFromDevice** cmdlet removes a primary user from the devices.

User device affinity is a method of associating a user with one or more specified devices in System Center 2012 Configuration Manager.

Parameters

-DeviceId<String[]>

Specifies an array of IDs of the devices.

Aliases	ResourceId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeviceName<String[]>

Specifies an array of names of the devices.

Aliases	ResourceName
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UserId<String>

Specifies the ID of a user.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UserName<String>

Specifies the name of the primary user that you want to disassociate from the specified devices.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Remove a primary user from a device

This command removes the association between the user that has the ID 206359374 and the device that has the ID 209846738.

```
PS C:\> Remove-CMUserAffinityFromDevice -DeviceId "209846738" -UserId "206359374"
```

Related topics

[Add-CMUserAffinityToDevice](#)

Remove-CMUserCollection

Remove-CMUserCollection

Removes user collections from the Configuration Manager hierarchy.

Syntax

Parameter Set: SearchByIdMandatory

```
Remove-CMUserCollection -CollectionId <String> [-Force] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByNameMandatory

```
Remove-CMUserCollection -Name <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Remove-CMUserCollection -InputObject <IResultObject> [-Force] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Detailed Description

The **Remove-CMUserCollection** cmdlet removes one or more user collections from the Microsoft System Center 2012 Configuration Manager hierarchy. You can specify the user collections by their name, ID, or an object that represents the collections.

Parameters

-CollectionId<String>

Specifies the IDs of the user collections to remove.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies an object that represents the user collections to remove.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies the names of the user collections to remove.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Remove a user collection

This command removes the user collection that has the ID 9990000D.

```
PS C:\> Remove-CMUserCollection -CollectionId "9990000D"
```

Related topics

[Get-CMUserCollection](#)

[Set-CMUserCollection](#)

[New-CMUserCollection](#)

[Import-CMUserCollection](#)

[Export-CMUserCollection](#)

Remove-CMUserCollectionDirectMembershipRule

Remove-CMUserCollectionDirectMembershipRule

Removes a direct membership rule from one or more user collection in the Configuration Manager hierarchy.

Syntax

Parameter Set: ByCollectionNameAndResourceName

```
Remove-CMUserCollectionDirectMembershipRule -CollectionName <String> -ResourceName <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByCollectionIdAndResourceId

```
Remove-CMUserCollectionDirectMembershipRule -CollectionId <String> -ResourceId <Int32> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByCollectionIdAndResourceName

```
Remove-CMUserCollectionDirectMembershipRule -CollectionId <String> -ResourceName <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByCollectionNameAndResourceId

```
Remove-CMUserCollectionDirectMembershipRule -CollectionName <String> -ResourceId <Int32> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByCollectionValueAndResourceId

```
Remove-CMUserCollectionDirectMembershipRule -Collection <IResultObject> -ResourceId <Int32> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByCollectionValueAndResourceName

```
Remove-CMUserCollectionDirectMembershipRule -Collection <IResultObject> -ResourceName <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Remove-CMUserCollectionDirectMembershipRule** cmdlet removes a direct rule from the specified collections. You can specify the collections by using their names, IDs, or by specifying an object that represents the collections.

For more information about collection rules in Microsoft System Center 2012 Configuration Manager, see [Introduction to Collections in Configuration Manager](http://go.microsoft.com/fwlink/p/?LinkID=259433) (<http://go.microsoft.com/fwlink/p/?LinkID=259433>) on TechNet.

Parameters

-Collection<IResultObject>

Specifies a Configuration Manager user collection object. To obtain a user collection object, use the **Get-CMUserCollection** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CollectionId<String>

Specifies an ID of a user collection.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CollectionName<String>

Specifies the name of a user collection.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ResourceId<Int32>

Specifies the ID of the direct rule to remove from the user collections.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ResourceName<String>

Specifies the name of the rule that you want to retrieve remove from the user collections.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

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

Example 1: Remove all direct membership rules from a user collection

This command removes all the direct membership rules of the user collection that has the ID CM0001A.

```
PS C:\> Remove-CMUserCollectionDirectMembershipRule -CollectionID "CM0001A" -ResourceId "12733"
```

Related topics

[Get-CMUserCollection](#)

[Add-CMUserCollectionDirectMembershipRule](#)

[Get-CMUserCollectionDirectMembershipRule](#)

[Remove-CMUserCollectionExcludeMembershipRule](#)

[Remove-CMUserCollectionIncludeMembershipRule](#)

[Remove-CMUserCollectionQueryMembershipRule](#)

Remove- CMUserCollectionExcludeMembershipRule

Remove-CMUserCollectionExcludeMembershipRule

Removes an exclude membership rule from one or more user collection in the Configuration Manager hierarchy.

Syntax

Parameter Set: ByCollectionNameAndExcludeCollectionName

```
Remove-CMUserCollectionExcludeMembershipRule -CollectionName <String> -ExcludeCollectionName <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByCollectionIdAndExcludeCollectionId

```
Remove-CMUserCollectionExcludeMembershipRule -CollectionId <String> -ExcludeCollectionId <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByCollectionIdAndExcludeCollectionName

```
Remove-CMUserCollectionExcludeMembershipRule -CollectionId <String> -ExcludeCollectionName <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByCollectionNameAndExcludeCollectionId

```
Remove-CMUserCollectionExcludeMembershipRule -CollectionName <String> -ExcludeCollectionId <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByCollectionValueAndExcludeCollectionId

```
Remove-CMUserCollectionExcludeMembershipRule -Collection <IResultObject> -ExcludeCollectionId <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByCollectionValueAndExcludeCollectionName

```
Remove-CMUserCollectionExcludeMembershipRule -Collection <IResultObject> -ExcludeCollectionName <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Remove-CMUserCollectionExcludeMembershipRule** cmdlet removes an exclude rule from the specified collections. You can specify the user collections by using their names, IDs, or by specifying an input object that represents the collections.

For more information about collection rules in Microsoft System Center 2012 Configuration Manager, see [Introduction to Collections in Configuration Manager](http://go.microsoft.com/fwlink/p/?LinkID=259433) (<http://go.microsoft.com/fwlink/p/?LinkID=259433>) on TechNet.

Parameters

-Collection<IResultObject>

Specifies a Configuration Manager user collection object. To obtain a user collection object, use the **Get-CMUserCollection** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CollectionId<String>

Specifies the IDs of a user collection.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CollectionName<String>

Specifies the name of a user collection.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-ExcludeCollectionId<String>

Specifies the ID of the collection whose members are excluded from the user collections.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ExcludeCollectionName<String>

Specifies the name of the collection whose members are excluded from the user collections.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

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

Example 1: Remove an exclude membership rule

This command removes the exclude membership rule that has the ID SMSDM001 from the user collection that has the ID in the 9990000D.

```
PS C:\> Remove-CMUserCollectionExcludeMembershipRule -CollectionId "9990000D" -  
ExcludeCollectionId "SMSDM001"
```

Related topics

[Add-CMUserCollectionExcludeMembershipRule](#)

[Get-CMUserCollectionExcludeMembershipRule](#)

[Get-CMUserCollection](#)

Remove- CMUserCollectionFromAdministrativeUser

Remove-CMUserCollectionFromAdministrativeUser

Removes a user collection from an administrative user in Configuration Manager.

Syntax

Parameter Set: RemoveUserCollectionFromAdminByName_Name

```
Remove-CMUserCollectionFromAdministrativeUser -AdministrativeUserName <String> -  
UserCollectionName <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: RemoveUserCollectionFromAdminById_Id

```
Remove-CMUserCollectionFromAdministrativeUser -AdministrativeUserId <Int32> -  
UserCollectionId <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: RemoveUserCollectionFromAdminById_Name

```
Remove-CMUserCollectionFromAdministrativeUser -AdministrativeUserName <String> -  
UserCollectionId <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: RemoveUserCollectionFromAdminById_Object

```
Remove-CMUserCollectionFromAdministrativeUser -AdministrativeUser <IResultObject> -  
UserCollectionId <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: RemoveUserCollectionFromAdminByName_Id

```
Remove-CMUserCollectionFromAdministrativeUser -AdministrativeUserId <Int32> -  
UserCollectionName <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: RemoveUserCollectionFromAdminByName_Object

```
Remove-CMUserCollectionFromAdministrativeUser -AdministrativeUser <IResultObject> -  
UserCollectionName <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: RemoveUserCollectionFromAdminByObject_Id

```
Remove-CMUserCollectionFromAdministrativeUser -AdministrativeUserId <Int32> -UserCollection  
<IResultObject> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: RemoveUserCollectionFromAdminByObject_Name

```
Remove-CMUserCollectionFromAdministrativeUser -AdministrativeUserName <String> -  
UserCollection <IResultObject> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: RemoveUserCollectionFromAdminByObject_Object

```
Remove-CMUserCollectionFromAdministrativeUser -AdministrativeUser <IResultObject> -  
UserCollection <IResultObject> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Remove-CMUserCollectionFromAdministrativeUser** cmdlet removes a user collection from an administrative user. In Microsoft System Center 2012 Configuration Manager, collections represent logical groupings of users or devices. Use collections to perform tasks such as managing settings or installing software updates.

Parameters

-AdministrativeUser<IResultObject>

Specifies a **CMUserCollection** object. To obtain a user collection object, use the **Get-CMUserCollection** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AdministrativeUserId<Int32>

Specifies a unique ID of an administrative group or user.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AdministrativeUserName<String>

Specifies a name of an administrative group or user.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UserCollection<IResultObject>

Specifies a **CMUserCollection** object. To obtain a user collection object, use the **Get-
CMUserCollection** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UserCollectionId<String>

Specifies a unique ID of a user collection associated with an administrative user.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UserCollectionName<String>

Specifies a name of a user collection associated with an administrative user.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Add a user collection to an administrative user

This command removes a user collection from an administrative user.

```
PS C:\> Remove-CMUserCollectionFromAdministrativeUser -AdministrativeUserName  
"TSQA\teamadmin" -UserCollectionName "All Users and User Groups"
```

Related topics

[Add-CMUserCollectionToAdministrativeUser](#)

Remove- CMUserCollectionFromDistributionPointGroup

Remove-CMUserCollectionFromDistributionPointGroup

Removes a user collection from a distribution point group in Configuration Manager.

Syntax

Parameter Set: SearchByIdMandatory

```
Remove-CMUserCollectionFromDistributionPointGroup -DistributionPointGroupId <String> -  
UserCollectionName <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByNameMandatory

```
Remove-CMUserCollectionFromDistributionPointGroup -DistributionPointGroupName <String> -  
UserCollectionName <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Remove-CMUserCollectionFromDistributionPointGroup -InputObject <IResultObject> -  
UserCollectionName <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Remove-CMUserCollectionFromDistributionPointGroup** cmdlet removes a user collection from a distribution point group in Microsoft System Center 2012 Configuration Manager. Distribution point groups contain distribution points that have a common purpose, such as supporting mobile device clients. A distribution point group simplifies tasks that involve the individual distribution points in a group.

Parameters

-DistributionPointGroupId<String>

Specifies an ID of a distribution group in Configuration Manager.

Aliases	GroupId
Required?	true
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DistributionPointGroupName<String>

Specifies a name of a distribution point group in Configuration Manager.

Aliases	GroupName
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a distribution point group object. To obtain a distribution point group object, use the **Get-CMDistributionPointGroup** cmdlet.

Aliases	none
Required?	true

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UserCollectionName<String>

Specifies a name of a user collection in Configuration Manager.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Remove a user collection by name

This command removes a user collection from a distribution point group by using a distribution point group ID and a user collection name.

```
PS C:\> Remove-CMUserCollectionFromDistributionPointGroup -DistributionPointGroupId "DPG-01" -UserCollectionName "Mobile Users"
```

Example 2: Remove a user collection by distribution point group name

This command removes a user collection from a distribution point group by using a distribution point group name and a user collection name.

```
PS C:\> Remove-CMUserCollectionFromDistributionPointGroup -DistributionPointGroupName "All Downtown" -UserCollectionName "Mobile Users"
```

Example 3: Remove a user collection by input object

This command removes a user collection from a distribution point group by using an input object and a user collection.

```
PS C:\> Remove-CMUserCollectionFromDistributionPointGroup -InputObject IRObjct-013 -UserCollectionName "All Remote"
```

Related topics

[Get-CMDistributionPointGroup](#)

[New-CMDistributionPointGroup](#)

[Set-CMDistributionPointGroup](#)

Remove- CMUserCollectionIncludeMembershipRule

Remove-CMUserCollectionIncludeMembershipRule

Removes an include membership rule from one or more user collection in the Configuration Manager hierarchy.

Syntax

Parameter Set: ByCollectionNameAndIncludeCollectionName

```
Remove-CMUserCollectionIncludeMembershipRule -CollectionName <String> -IncludeCollectionName <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByCollectionIdAndIncludeCollectionId

```
Remove-CMUserCollectionIncludeMembershipRule -CollectionId <String> -IncludeCollectionId <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByCollectionIdAndIncludeCollectionName

```
Remove-CMUserCollectionIncludeMembershipRule -CollectionId <String> -IncludeCollectionName <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByCollectionNameAndIncludeCollectionId

```
Remove-CMUserCollectionIncludeMembershipRule -CollectionName <String> -IncludeCollectionId <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByCollectionValueAndIncludeCollectionId

```
Remove-CMUserCollectionIncludeMembershipRule -Collection <IResultObject> -IncludeCollectionId <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByCollectionValueAndIncludeCollectionName

```
Remove-CMUserCollectionIncludeMembershipRule -Collection <IResultObject> -IncludeCollectionName <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Remove-CMUserCollectionIncludeMembershipRule** cmdlet removes an include rule from the specified collections. You can specify the user collections by using their names, IDs, or by specifying an input object that represents the collections.

For more information about collection rules in Microsoft System Center 2012 Configuration Manager, see [Introduction to Collections in Configuration Manager](http://go.microsoft.com/fwlink/p/?LinkID=259433) (<http://go.microsoft.com/fwlink/p/?LinkID=259433>) on TechNet.

Parameters

-Collection<IResultObject>

Specifies a Configuration Manager user collection object. To obtain a user collection object, use the **Get-CMUserCollection** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CollectionId<String>

Specifies the ID of the user collection.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CollectionName<String>

Specifies the name of the user collection.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-IncludeCollectionId<String>

Specifies the ID for the collection whose members are included in the user collections.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-IncludeCollectionName<String>

Specifies the name for the collection whose members are included in the user collections.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

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

Example 1: Remove an include membership rule from a user collection

This command removes the include membership rule that has the ID SMSDM001 from the user collection that has the ID 9990000D.

```
PS C:\> Remove-CMUserCollectionIncludeMembershipRule -CollectionId "9990000D" -  
IncludeCollectionId "SMSDM001"
```

Related topics

[Get-CMUserCollection](#)

[Add-CMUserCollectionIncludeMembershipRule](#)

[Get-CMUserCollectionIncludeMembershipRule](#)

Remove- CMUserCollectionQueryMembershipRule

Remove-CMUserCollectionQueryMembershipRule

Removes a query membership rule from one or more user collection in the Configuration Manager hierarchy.

Syntax

Parameter Set: ByCollectionName

```
Remove-CMUserCollectionQueryMembershipRule -CollectionName <String> -RuleName <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByCollectionId

```
Remove-CMUserCollectionQueryMembershipRule -CollectionId <String> -RuleName <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByCollectionValue

```
Remove-CMUserCollectionQueryMembershipRule -Collection <IResultObject> -RuleName <String> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Remove-CMUserCollectionQueryMembershipRule** cmdlet removes a query rule from the specified user collections. You can specify the user collections by using their names, IDs, or by specifying an input object that represents the collections.

For more information about membership rules in Microsoft System Center 2012 Configuration Manager, see [Introduction to Collections in Configuration Manager](http://go.microsoft.com/fwlink/p/?LinkID=259433) (<http://go.microsoft.com/fwlink/p/?LinkID=259433>) on TechNet.

Parameters

-Collection<IResultObject>

Specifies a Configuration Manager user collection object. To obtain a user collection object, use the **Get-CMUserCollection** cmdlet.

Aliases	none
---------	------

Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CollectionId<String>

Specifies the ID of the user collection.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CollectionName<String>

Specifies the name of the user collection.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Force

Performs the action without a confirmation message.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RuleName<String>

Specifies the name of the query rule to remove from the user collections.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Remove a rule from a collection by using the collection name

This command removes the rule named Remote Users by Domain from the collection named Remote Users.

```
PS C:\> Remove-CMUserCollectionQueryMembershipRule -CollectionName "Remote Users" -RuleName "Remote Users by Domain"
```

Related topics

[Get-CMUserCollection](#)

[Add-CMUserCollectionQueryMembershipRule](#)

[Get-CMUserCollectionQueryMembershipRule](#)

Remove-CMUserDataAndProfileConfigurationItem

Remove-CMUserDataAndProfileConfigurationItem

Removes user data and profile configuration items.

Syntax

Parameter Set: SearchByNameMandatory

```
Remove-CMUserDataAndProfileConfigurationItem -Name <String[]> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Remove-CMUserDataAndProfileConfigurationItem -Id <String[]> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Remove-CMUserDataAndProfileConfigurationItem -InputObject <IResultObject> [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Remove-CMUserDataAndProfileConfigurationItem** cmdlet removes one or more user data and profile configuration items. You must remove all references to a configuration item before you can delete the configuration item.

After you remove a user data and profile configuration item, you cannot use the configuration item to manage folder redirection, offline files, and roaming profiles on computers that run Windows® 8 for the user collections where you deployed the user data and profile configuration item.

Parameters

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Id<String[]>

Specifies an array of IDs of user data and profile configuration items.

Aliases	Ciid
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a **CMUserDataAndProfileConfigurationItem** object. To obtain a **CMUserDataAndProfileConfigurationItem** object, use the **Get-
CMUserDataAndProfileConfigurationItem** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String[]>

Specifies an array of names of user data and profile configuration items.

Aliases	LocalizedDisplayName
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Remove a user data and profile configuration item

This command removes the user data and profile configuration item named UDPConfig02.

```
PS C:\> Remove-CMUserDataAndProfileConfigurationItem -Name "UDPConfig02"
```

Related topics

[Copy-CMUserDataAndProfileConfigurationItem](#)

[Get-CMUserDataAndProfileConfigurationItem](#)

[New-CMUserDataAndProfileConfigurationItem](#)

[Set-CMUserDataAndProfileConfigurationItem](#)

Remove-CMVhd

Remove-CMVhd

Removes VHD images.

Syntax

Parameter Set: SearchByIdMandatory

```
Remove-CMVhd -Id <String[]> [-Force] [-SecuredScopeNames <String> ] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByNameMandatory

```
Remove-CMVhd -Name <String> [-Force] [-SecuredScopeNames <String> ] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Remove-CMVhd -InputObject <IResultObject> [-Force] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Detailed Description

The **Remove-CMVhd** cmdlet removes one or more virtual hard disk (VHD) images from the operating system deployment feature. This cmdlet does not delete VHD images.

Parameters

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Id<String[]>

Specifies an array of IDs of VHD images.

Aliases	Packageld
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a VHD image object. To obtain a VHD image object, use the **Get-CMVhd** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies the name of a VHD image.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SecuredScopeNames<String>

Specifies names of security scopes. A security scope name can be Default or the name of a custom-created security scope.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

[Get-CMVhd](#)

[New-CMVhd](#)

[Set-CMVhd](#)

Remove-CMVpnProfileConfigurationItem

Remove-CMVpnProfileConfigurationItem

Removes a VPN profile.

Syntax

Parameter Set: SearchByNameMandatory

```
Remove-CMVpnProfileConfigurationItem -Name <String[]> [-Force] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Remove-CMVpnProfileConfigurationItem -Id <String[]> [-Force] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Remove-CMVpnProfileConfigurationItem -InputObject <IResultObject> [-Force] [-Confirm] [-  
WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Remove-CMVpnProfileConfigurationItem** cmdlet removes a virtual private network (VPN) profile. Clients download the updated profile on the regular schedule.

Parameters

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Id<String[]>

Specifies an array of IDs of VPN profile objects.

Aliases	CId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a VPN profile object. To obtain a VPN profile object, use the **Get-CMVpnProfileConfigurationItem** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String[]>

Specifies an array of names of VPN profiles.

Aliases	LocalizedDisplayName
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

[Copy-CMVpnProfileConfigurationItem](#)

[Get-CMVpnProfileConfigurationItem](#)

[New-CMVpnProfileConfigurationItem](#)

[Set-CMVpnProfileConfigurationItem](#)

Remove-CMWindowsFirewallPolicy

Remove-CMWindowsFirewallPolicy

Removes Windows Firewall policies for Endpoint Protection.

Syntax

Parameter Set: SearchByNameMandatory

```
Remove-CMWindowsFirewallPolicy -Name <String[]> [-Force] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Remove-CMWindowsFirewallPolicy -Id <String[]> [-Force] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Remove-CMWindowsFirewallPolicy -InputObject <IResultObject> [-Force] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Detailed Description

The **Remove-CMWindowsFirewallPolicy** cmdlet removes one or more Windows Firewall policies Microsoft System Center 2012 Endpoint Protection in Microsoft System Center 2012 Configuration Manager. When you remove a Windows Firewall policy, System Center 2012 Configuration Manager removes the policy from the computers on which you deployed the policy.

Parameters

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-Id<String[]>

Specifies an array of IDs of Windows Firewall policies.

Aliases	CIId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a **CMWindowsFirewallPolicy** object. To obtain a **CMWindowsFirewallPolicy** object, use the **Get-CMWindowsFirewallPolicy** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String[]>

Specifies an array of Windows Firewall policy names.

Aliases	LocalizedDisplayName
Required?	true
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Remove a Windows Firewall policy by using a name

This command removes the Windows Firewall policy that has the ID WFPContoso01.

```
PS C:\> Remove-CMWindowsFirewallPolicy -Name "WFPContoso01"
```

Example 2: Remove a Windows Firewall policy by using an object variable

The first command gets the **CMWindowsFirewallPolicy** object that has the ID 16777568 and stores it in the \$WFPobj variable.

The second command removes the Windows Firewall policy stored in the \$WFPobj variable.

```
PS C:\> $WFPobj=Get-CMWindowsFirewallPolicy -Id "16777568"
```

```
PS C:\> Remove-CMWindowsFirewallPolicy -InputObject $WFPobj
```

Related topics

[Get-CMWindowsFirewallPolicy](#)

[New-CMWindowsFirewallPolicy](#)

[Set-CMWindowsFirewallPolicy](#)

Remove-CMWirelessProfileConfigurationItem

Remove-CMWirelessProfileConfigurationItem

Removes wireless profiles.

Syntax

Parameter Set: SearchByNameMandatory

```
Remove-CMWirelessProfileConfigurationItem -Name <String[]> [-Force] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Remove-CMWirelessProfileConfigurationItem -Id <String[]> [-Force] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Remove-CMWirelessProfileConfigurationItem -InputObject <IResultObject> [-Force] [-Confirm]  
[-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Remove-CMWirelessProfileConfigurationItem** cmdlet removes wireless profiles. Clients download the updated profile on the regular schedule.

Parameters

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Id<String[]>

Specifies an array of IDs of wireless profiles.

Aliases	CId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a wireless profile object. To obtain a wireless profile object, use the **Get-CMWirelessProfileConfigurationItem** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String[]>

Specifies an array of names of wireless profiles.

Aliases	LocalizedDisplayName
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

[Copy-CMWirelessProfileConfigurationItem](#)

[Get-CMWirelessProfileConfigurationItem](#)

[New-CMWirelessProfileConfigurationItem](#)

[Set-CMWirelessProfileConfigurationItem](#)

Resolve-CMInventoriedSoftwareConflict

Resolve-CMInventoriedSoftwareConflict

Resolves a conflict in Configuration Manager software inventory information.

Syntax

Parameter Set: SearchByIdMandatory

```
Resolve-CMInventoriedSoftwareConflict -Id <String[]> -RevertLocalEdit <Boolean> [-Confirm]  
[-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Resolve-CMInventoriedSoftwareConflict** cmdlet resolves a conflict in Microsoft System Center 2012 Configuration Manager software inventory information.

When System Center 2012 Configuration Manager receives updated information about software that is part of the software inventory, that information may conflict with your local settings. You can resolve a conflict by keeping your local inventory information or updating to the new information.

Parameters

-Id<String[]>

Specifies an array of IDs for conflicts in software inventory.

Aliases	SoftwareKey
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RevertLocalEdit<Boolean>

Indicates whether this cmdlet keeps the current inventory information for the conflict or updates that information. If this parameter is \$True, the cmdlet keeps current, local information. If this parameter is \$False, the cmdlet replaces conflicting information with updated information.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Resolve a software conflict and keep local inventory information

This command resolves a software conflict that has the specified ID. The command keeps the current, local version of the conflicting information.

```
PS C:\> Resolve-CMInventoriedSoftwareConflict -Id "SMS0001" -RevertLocalEdit $True
```

Related topics

[Get-CMSoftwareInventory](#)

[Set-CMSoftwareInventory](#)

[Undo-CMSoftwareInventory](#)

Restore-CMApplicationRevisionHistory

Restore-CMApplicationRevisionHistory

Restores a previous version of a Configuration Manager application from the application revision history.

Syntax

Parameter Set: SearchByNameMandatory

```
Restore-CMApplicationRevisionHistory -Name <String[]> -Revision <Int32> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Restore-CMApplicationRevisionHistory -Id <String[]> -Revision <Int32> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Restore-CMApplicationRevisionHistory -InputObject <IResultObject> -Revision <Int32> [-  
Confirm] [-WhatIf] [  
<CommonParameters>]
```

Detailed Description

The **Restore-CMApplicationRevisionHistory** cmdlet restores a previous version of a Microsoft System Center 2012 Configuration Manager application. You can use the revision history that System Center 2012 Configuration Manager creates and maintains for each application to choose the version of the application that you want to restore.

If you restore an application version from the history, System Center 2012 Configuration Manager might automatically replace currently installed copies of the application the next time it evaluates the deployment schedule. For more control over application replacement, create a new application that supersedes the application that you want to replace, and then deploy this application to the required collection.

Parameters

-Id<String[]>

Specifies an array of IDs of application revision histories.

Aliases	CId
---------	-----

Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies an object that contains an application revision history. To obtain this object, use the [Get-CMApplicationRevisionHistory](#) cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String[]>

Specifies an array of names of application revision histories.

Aliases	LocalizedDisplayName
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Revision<Int32>

Specifies the version number of the application revision that you restore.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

[Get-CMApplicationRevisionHistory](#)

[Remove-CMApplicationRevisionHistory](#)

Resume-CMApplication

Resume-CMApplication

Resumes an application in Configuration Manager.

Syntax

Parameter Set: SearchByIdMandatory

```
Resume-CMApplication -Id <String[]> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByNameMandatory

```
Resume-CMApplication -Name <String[]> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Resume-CMApplication -InputObject <IResultObject> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Resume-CMApplication** cmdlet resumes an application. If the application was suspended, clients can now download the application.

Parameters

-Id<String[]>

Specifies an array of application IDs.

Aliases	CIId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies an application object. To obtain an application object, use the **Get-CMApplication** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String[]>

Specifies an array of application names.

Aliases	LocalizedDisplayName
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Resume an application by using ID

This command resumes an application using the identifier.

```
PS C:\> Resume-CMApplication -Id "16781415"
```

Related topics

[Export-CMApplication](#)

[Get-CMApplication](#)

[Import-CMApplication](#)

[New-CMApplication](#)

[Remove-CMApplication](#)

[Set-CMApplication](#)

[Suspend-CMApplication](#)

Save-CMDatabaseReplicationDiagnostic

Save-CMDatabaseReplicationDiagnostic

Saves database replication diagnostic information for Configuration Manager in a file.

Syntax

Parameter Set: SearchBySiteMandatory

```
Save-CMDatabaseReplicationDiagnostic -ChildSiteCode <String> -FileName <String> -  
ParentSiteCode <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Save-CMDatabaseReplicationDiagnostic** cmdlet saves diagnostic information for database replication issues for Microsoft System Center 2012 Configuration Manager in a specified file. This cmdlet runs diagnostics for a link between a parent and a child site databases. You can specify sites by either name or site code, but you cannot specify one site by name and the other by site code.

System Center 2012 Configuration Manager database replication transfers data and merges changes made in a site database with the information stored in other sites in the System Center 2012 Configuration Manager site hierarchy so that all sites share the same information. System Center 2012 Configuration Manager configures database replication automatically between a parent and child site. Diagnostics identify problems in database replication.

Parameters

-ChildSiteCode<String>

Specifies a site code for a Configuration Manager site. This is the child site.

Aliases	Site2
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-FileName<String>

Specifies a file name. This cmdlet saves database diagnostic information for database replication to this file.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ParentSiteCode<String>

Specifies a site code for a Configuration Manager site. This is the parent site.

Aliases	Site1
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Save database replication diagnostic

This command saves database replication diagnostics in a file named CCB_CC2_Diagnostics.csv. The command specifies a parent and child site by using site codes.

```
PS C:\> Save-CMDatabaseReplicationDiagnostic -ChildSiteCode "CC2" -FileName  
"D:\Diagnostics\CCB_CC2_Diagnostics.csv" -ParentSiteCode "CCB"
```

Related topics

[Get-CMDataBaseReplicationStatus](#)

[Get-CMDatabaseReplicationLinkProperty](#)

Save-CMEndpointProtectionDefinition

Save-CMEndpointProtectionDefinition

Saves an Endpoint Protection definition.

Syntax

Parameter Set: SearchByNameMandatory

```
Save-CMEndpointProtectionDefinition -DeviceCollectionName <String> [-Device <IResultObject> ] [-DeviceId <String> ] [-DeviceName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Save-CMEndpointProtectionDefinition -DeviceCollectionId <String> [-Device <IResultObject> ] [-DeviceId <String> ] [-DeviceName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Save-CMEndpointProtectionDefinition -DeviceCollection <IResultObject> [-Device <IResultObject> ] [-DeviceId <String> ] [-DeviceName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Save-CMEndpointProtectionDefinition** cmdlet saves a System Center 2012 Endpoint Protection definition in Microsoft System Center 2012 Configuration Manager. Endpoint Protection definitions contain anti-malware policies and settings for Windows Firewall that you can apply to specific groups of computers.

For more information about Endpoint Protection, see [Endpoint Protection in Configuration Manager](http://go.microsoft.com/fwlink/?linkid=268427) (<http://go.microsoft.com/fwlink/?linkid=268427>) on TechNet.

Parameters

-Device<IResultObject>

Specifies a device object in Configuration Manager. To obtain a device object, use the **CM-GetDevice** cmdlet. This object identifies the device to which you save the Endpoint Protection definition.

Aliases	none
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeviceCollection<IResultObject>

Specifies a device collection object in Configuration Manager. To obtain a device collection object, use the **CM-GetDeviceCollection** cmdlet. This object identifies the device collection to which you save the Endpoint Protection definition.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeviceCollectionId<String>

Specifies an ID for a Configuration Manager device collection to which you add the Endpoint Protection definition.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeviceCollectionName<String>

Specifies a name for a Configuration Manager device collection to which you add the Endpoint Protection definition.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeviceId<String>

Specifies the ID of a Configuration Manager device to which you add the Endpoint Protection definition.

Aliases	ResourceID
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeviceName<String>

Specifies the name of a Configuration Manager device to which you save the Endpoint Protection definition.

Aliases	Name
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Save an Endpoint Protection definition by using a device collection name

This command saves the Endpoint Protection definition to the devices in the device collection named NA-Client-Devices.

```
PS C:\> Save-CMEndpointProtectionDefinition -DeviceCollectionName "NA-Client-Devices"
```

Related topics

[Add-CMEndpointProtectionPoint](#)

[Get-CMEndpointProtectionPoint](#)

[Remove-CMEndpointProtectionPoint](#)

Save-CMSoftwareUpdate

Save-CMSoftwareUpdate

Saves Configuration Manager software updates.

Syntax

Parameter Set: SearchByNameMandatory

```
Save-CMSoftwareUpdate -DeploymentPackageName <String> -SoftwareUpdateName <String[]> [-Location <String> ] [-SoftwareUpdateLanguage <String[]> ] [-Confirm] [-WhatIf] [<CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Save-CMSoftwareUpdate -DeploymentPackageName <String> -SoftwareUpdateId <String[]> [-Location <String> ] [-SoftwareUpdateLanguage <String[]> ] [-Confirm] [-WhatIf] [<CommonParameters>]
```

Parameter Set: SearchByIdMandatory_UpdateGroup

```
Save-CMSoftwareUpdate -DeploymentPackageName <String> -SoftwareUpdateGroupId <String[]> [-Location <String> ] [-SoftwareUpdateLanguage <String[]> ] [-Confirm] [-WhatIf] [<CommonParameters>]
```

Parameter Set: SearchByNameMandatory_UpdateGroup

```
Save-CMSoftwareUpdate -DeploymentPackageName <String> -SoftwareUpdateGroupName <String[]> [-Location <String> ] [-SoftwareUpdateLanguage <String[]> ] [-Confirm] [-WhatIf] [<CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Save-CMSoftwareUpdate -DeploymentPackageName <String> -SoftwareUpdate <IResultObject> [-Location <String> ] [-SoftwareUpdateLanguage <String[]> ] [-Confirm] [-WhatIf] [<CommonParameters>]
```

Parameter Set: SearchByValueMandatory_UpdateGroup

```
Save-CMSoftwareUpdate -DeploymentPackageName <String> -SoftwareUpdateGroup <IResultObject> [-Location <String> ] [-SoftwareUpdateLanguage <String[]> ] [-Confirm] [-WhatIf] [<CommonParameters>]
```

Detailed Description

The **Save-CMSoftwareUpdate** cmdlet saves one or more software updates in Microsoft System Center 2012 Configuration Manager.

You can specify one or more software updates associated with deployment packages. You can also specify the location to save the updates.

You can also specify the language of the software updates. Languages determine which summary details a software update synchronizes and the file languages to be downloaded for software updates.

Parameters

-DeploymentPackageName<String>

Specifies a name for a deployment package.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Location<String>

Specifies a location.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SoftwareUpdate<IResultObject>

Specifies a software update object. To obtain a software update object, use the **Get-CMSoftwareUpdate** cmdlet.

Aliases	none
Required?	true
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SoftwareUpdateGroup<IResultObject>

Specifies a software update group object. To obtain a software update group object, use the **Get-CMSoftwareUpdateGroup** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SoftwareUpdateGroupId<String[]>

Specifies an array of IDs of software groups.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SoftwareUpdateGroupName<String[]>

Specifies an array of names of software groups.

Aliases	none
Required?	true

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SoftwareUpdateId<String[]>

Specifies an array of IDs of software updates.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SoftwareUpdateLanguage<String[]>

Specifies an array of software update languages.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SoftwareUpdateName<String[]>

Specifies an array of names of software updates.

Aliases	LocalizedDisplayName
Required?	true

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Save a software update by using an ID

This command saves the software update that has the ID 16777373, which is part of the deployment package named PackageDeploy22 in the specified location.

```
PS C:\> Save-CMSoftwareUpdate -DeploymentPackageName "PackageDeploy22" -SoftwareUpdateId "16777373" -Location "\\ContosoUpdates\PackageDeploy22"
```

Example 2: Save a software update that has a specified language

This command saves the named software update that is part of package named PackageDeploy24 in the specified location. The command also specifies English as the language for the software updates.

```
PS C:\> Save-CMSoftwareUpdate -DeploymentPackageName "PackageDeploy24" -SoftwareUpdateGroupName "Accounting package updates" -Location "\\ContosoUpdates\PackageDeploy24" -SoftwareUpdateLanguage English
```

Related topics

[Set-CMSoftwareUpdate](#)

[Sync-CMSoftwareUpdate](#)

[Get-CMSoftwareUpdate](#)

[Get-CMSoftwareUpdateGroup](#)

Send-CMAssetIntelligenceCatalogUpdateRequest

Send-CMAssetIntelligenceCatalogUpdateRequest

Requests a catalog update for uncategorized software titles.

Syntax

Parameter Set: SearchByNameMandatory

```
Send-CMAssetIntelligenceCatalogUpdateRequest -Name <String[]> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Send-CMAssetIntelligenceCatalogUpdateRequest -Id <String[]> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Detailed Description

The **Send-CMAssetIntelligenceCatalogUpdateRequest** cmdlet requests an update of the Asset Intelligence catalog for software title categorization from System Center Online. You can request an update for catalog items or software categories in the Asset Intelligence catalog.

You can also use the [Sync-CMAssetIntelligenceCatalog](#) cmdlet to synchronize the local Asset Intelligence catalog with System Center Online to retrieve the latest software title categorization.

Parameters

-Id<String[]>

Specifies an array of IDs of Asset Intelligence catalog items.

Aliases	SoftwareKey
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-Name<String[]>

Specifies an array of names of software categories in the Asset Intelligence catalog.

Aliases	CommonName
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	true

-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

Example 1: Request an update for a software category

This command requests an update of the Asset Intelligence catalog for the software category named Browsers.

```
PS C:\> Send-CMAAssetIntelligenceCatalogUpdateRequest -Name "Browsers"
```

Related topics

[Get-CMAAssetIntelligenceCatalogItem](#)

[Sync-CMAAssetIntelligenceCatalog](#)

Set-CMAccessAccount

Set-CMAccessAccount

Modifies the properties of an access account.

Syntax

Parameter Set: SearchByApplicationName

```
Set-CMAccessAccount -AccountType {Administrator | Guest | User | WindowsGroup | WindowsUser}
-ApplicationName <String> [-Access {Change | FullControl | NoAccess | Read} ] [-UserName
<String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByApplication

```
Set-CMAccessAccount -AccountType {Administrator | Guest | User | WindowsGroup | WindowsUser}
-Application <IResultObject> [-Access {Change | FullControl | NoAccess | Read} ] [-UserName
<String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByApplicationId

```
Set-CMAccessAccount -AccountType {Administrator | Guest | User | WindowsGroup | WindowsUser}
-ApplicationId <String> [-Access {Change | FullControl | NoAccess | Read} ] [-UserName
<String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByBootImage

```
Set-CMAccessAccount -AccountType {Administrator | Guest | User | WindowsGroup | WindowsUser}
-BootImage <IResultObject> [-Access {Change | FullControl | NoAccess | Read} ] [-UserName
<String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByBootImageId

```
Set-CMAccessAccount -AccountType {Administrator | Guest | User | WindowsGroup | WindowsUser}
-BootImageId <String> [-Access {Change | FullControl | NoAccess | Read} ] [-UserName
<String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByBootImageName

```
Set-CMAccessAccount -AccountType {Administrator | Guest | User | WindowsGroup | WindowsUser}
-BootImageName <String> [-Access {Change | FullControl | NoAccess | Read} ] [-UserName
<String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByDriverPackage

```
Set-CMAccessAccount -AccountType {Administrator | Guest | User | WindowsGroup | WindowsUser}
-DriverPackage <IResultObject> [-Access {Change | FullControl | NoAccess | Read} ] [-
UserName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByDriverPackageId

```
Set-CMAccessAccount -AccountType {Administrator | Guest | User | WindowsGroup | WindowsUser}
-DriverPackageId <String> [-Access {Change | FullControl | NoAccess | Read} ] [-UserName
<String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByDriverPackageName

Set-CMAccessAccount -AccountType {Administrator | Guest | User | WindowsGroup | WindowsUser}
-DriverPackageName <String> [-Access {Change | FullControl | NoAccess | Read}] [-UserName
<String>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SearchByOSImage

Set-CMAccessAccount -AccountType {Administrator | Guest | User | WindowsGroup | WindowsUser}
-OperatingSystemImage <IResultObject> [-Access {Change | FullControl | NoAccess | Read}] [-
UserName <String>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SearchByOSImageId

Set-CMAccessAccount -AccountType {Administrator | Guest | User | WindowsGroup | WindowsUser}
-OperatingSystemImageId <String> [-Access {Change | FullControl | NoAccess | Read}] [-
UserName <String>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SearchByOSImageName

Set-CMAccessAccount -AccountType {Administrator | Guest | User | WindowsGroup | WindowsUser}
-OperatingSystemImageName <String> [-Access {Change | FullControl | NoAccess | Read}] [-
UserName <String>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SearchByOSInstaller

Set-CMAccessAccount -AccountType {Administrator | Guest | User | WindowsGroup | WindowsUser}
-OperatingSystemInstaller <IResultObject> [-Access {Change | FullControl | NoAccess | Read}
] [-UserName <String>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SearchByOSInstallerId

Set-CMAccessAccount -AccountType {Administrator | Guest | User | WindowsGroup | WindowsUser}
-OperatingSystemInstallerId <String> [-Access {Change | FullControl | NoAccess | Read}] [-
UserName <String>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SearchByOSInstallerName

Set-CMAccessAccount -AccountType {Administrator | Guest | User | WindowsGroup | WindowsUser}
-OperatingSystemInstallerName <String> [-Access {Change | FullControl | NoAccess | Read}]
[-UserName <String>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SearchByPackage

Set-CMAccessAccount -AccountType {Administrator | Guest | User | WindowsGroup | WindowsUser}
-Package <IResultObject> [-Access {Change | FullControl | NoAccess | Read}] [-UserName
<String>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SearchByPackageId

Set-CMAccessAccount -AccountType {Administrator | Guest | User | WindowsGroup | WindowsUser}
-PackageId <String> [-Access {Change | FullControl | NoAccess | Read}] [-UserName <String>
] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SearchByPackageName

Set-CMAccessAccount -AccountType {Administrator | Guest | User | WindowsGroup | WindowsUser}
-PackageName <String> [-Access {Change | FullControl | NoAccess | Read}] [-UserName
<String>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SearchBySoftwareUpdateDeploymentPackage

Set-CMAccessAccount -AccountType {Administrator | Guest | User | WindowsGroup | WindowsUser}

```
-SoftwareUpdateDeploymentPackage <IResultObject> [-Access {Change | FullControl | NoAccess | Read} ] [-UserName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchBySoftwareUpdateDeploymentPackageId

```
Set-CMAccessAccount -AccountType {Administrator | Guest | User | WindowsGroup | WindowsUser} -SoftwareUpdateDeploymentPackageId <String> [-Access {Change | FullControl | NoAccess | Read} ] [-UserName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchBySoftwareUpdateDeploymentPackageName

```
Set-CMAccessAccount -AccountType {Administrator | Guest | User | WindowsGroup | WindowsUser} -SoftwareUpdateDeploymentPackageName <String> [-Access {Change | FullControl | NoAccess | Read} ] [-UserName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Set-CMAccessAccount** cmdlet modifies the properties of an access account. You can add users or groups to the access account and change the level of permissions to objects to which they have permissions.

An access account is a list of users or groups that can access an established service or application that is located on a distribution point. For example, members in the Software Update Point Connection Access Account can access two services to manage software updates: Windows Server Update Services (WSUS) and WSUS Synchronization Manager.

Parameters

-Access<AccessRight>

Specifies the access rights that are associated with an access account. Valid values are: No Access, Read, Change, and Full Control.

The acceptable values for this parameter are:

Change	
FullControl	
NoAccess	
Read	

Aliases	none
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AccountType<AccessAccountType>

Specifies an account type. Valid values are: Guest, User, WindowsGroup, and WindowsUser.

The acceptable values for this parameter are:

Administrator	
Guest	
User	
WindowsGroup	
WindowsUser	

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Application<IResultObject>

Specifies a deployed application object. You can get an application object by using the **Get-CAApplication** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ApplicationId<String>

Specifies the ID of an application.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ApplicationName<String>

Specifies the name of an application.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-BootImage<IResultObject>

Specifies a boot image object. A boot image object contains the Windows files that are required to prepare a computer for the installation of an operating system. You can get a boot image object by using the **Get-CMBootImage** cmdlet.

Aliases	none
Required?	true

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-BootImageId<String>

Specifies the ID of a boot image.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-BootImageName<String>

Specifies the name of a boot image.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DriverPackage<IResultObject>

Specifies a driver package object. A driver package object specifies a group of hardware drivers that are required to install an operating system. You can get a driver package object by using the **Get-CMDriverPackage** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DriverPackageId<String>

Specifies the ID of a driver package.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DriverPackageName<String>

Specifies the name of a driver package.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OperatingSystemImage<IResultObject>

Specifies an operating system image object. An operating system image object contains the Windows files that compose a complete Windows installation. You can get an operating system image object by using the **Get-CMOperatingSystemImage** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OperatingSystemImageId<String>

Specifies the ID of an operating system image.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OperatingSystemImageName<String>

Specifies the name of an operating system image.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OperatingSystemInstaller<IResultObject>

Specifies an operating system installer object. An operating system installer object contains the Windows files that are required to prepare a computer for the installation of an operating system. To obtain an operating system installer object, use the **Get-CMOperatingSystemInstaller** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OperatingSystemInstallerId<String>

Specifies the ID of an operating system installer.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OperatingSystemInstallerName<String>

Specifies the name of an operating system installer.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Package<IResultObject>

Specifies a deployed software script or program object. You can get a package by using the **Get-CMPackage** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PackageId<String>

Specifies the ID of a deployed software script or program.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PackageName<String>

Specifies the name of a deployed software script or program.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-SoftwareUpdateDeploymentPackage<IResultObject>

Specifies a deployed software update object. You can get a software update deployment object by using the **Get-CMSoftwareUpdateDeploymentPackage** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SoftwareUpdateDeploymentPackageId<String>

Specifies the ID of a deployed software update.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SoftwareUpdateDeploymentPackageName<String>

Specifies the name of a deployed software update.

Aliases	none
Required?	true
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UserName<String>

Specifies a Windows user account name in *domain\user* format.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Change access to a package by using the package name

In this example, the first command gets the package name and stores it in the variable \$Name.

The second command sets access permissions for the user to the package to Read. You must confirm the action before the command performs it.

```
PS C:\> $Name = Get-CMAccessAccount -PackageName "Configuration Manager Client Package"
PS C:\> Set-CMAccessAccount -PackageName $Name -Type User -UserName "CONTOSO\PFuller" -
Access Read -Confirm
```

Related topics

[Get-CMAccessAccount](#)

[New-CMAccessAccount](#)

[Remove-CMAccessAccount](#)

[Get-CMApplication](#)

[Get-CMBootImage](#)

[Get-CMDriverPackage](#)

[Get-CMOperatingSystemImage](#)

[Get-CMOperatingSystemInstaller](#)

[Get-CMPackage](#)

[Get-CMSoftwareUpdateDeploymentPackage](#)

Set-CMAccount

Set-CMAccount

Sets a Configuration Manager user account.

Syntax

Parameter Set: SetAccountByName

```
Set-CMAccount -Name <String> -Password <SecureString> -SiteCode <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetAccountByObject

```
Set-CMAccount -InputObject <IResultObject> -Password <SecureString> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Set-CMAccount** cmdlet sets a user account in Microsoft System Center 2012 Configuration Manager. A **CMAccount** is a user account that Configuration Manager uses to connect to various system and network resources. For more information about user accounts, see [Technical Reference for Accounts Used in Configuration Manager](http://go.microsoft.com/fwlink/?LinkID=248317) (<http://go.microsoft.com/fwlink/?LinkID=248317>) on TechNet.

Parameters

-InputObject<IResultObject>

Specifies a user account object. You can get a user account object by using the **Get-CMAccount** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies a name for the user account.

Aliases	UserName
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Password<SecureString>

Specifies a secure string that contains the password for the user account.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteCode<String>

Specifies a Configuration Manager site code.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Set an account by using name and password

The first command creates a variable as a secure string.

The second command creates another variable as a secure string.

The third command sets the password for the account.

```
PS C:\> $Secure = Read-Host -AsSecureString
PS C:\> $ConfirmSecure = Read-Host -AsSecureString
PS C:\> Set-CMAccount -Name " TSQA\PFuller" -Password $Secure -ConfirmPassword
$ConfirmSecure -SiteCode "CM2"
```

Related topics

[Get-CMAccount](#)

[New-CMAccount](#)

[Remove-CMAccount](#)

Set-CMActiveDirectoryForest

Set-CMActiveDirectoryForest

Changes Active Directory forest properties in Configuration Manager.

Syntax

Parameter Set: SetById

```
Set-CMActiveDirectoryForest -Id <String[]> [-Description <String> ] [-EnableDiscovery <Boolean> ] [-PublishingPath <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetByFQDN

```
Set-CMActiveDirectoryForest -ForestFqdn <String> [-Description <String> ] [-EnableDiscovery <Boolean> ] [-PublishingPath <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetValue

```
Set-CMActiveDirectoryForest -InputObject <IResultObject> [-Description <String> ] [-EnableDiscovery <Boolean> ] [-PublishingPath <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Set-CMActiveDirectoryForest** cmdlet changes values for an Active Directory forest object in Microsoft System Center 2012 Configuration Manager. You can edit the description, enable or disable discovery, and specify a fully qualified domain name (FQDN) and publishing path. You can specify an Active Directory forest object by ID or FQDN, or you can supply the Active Directory forest object itself. Active Directory Forest Discovery requires a global account to discover or publish to untrusted forests.

Parameters

-Description<String>

Specifies a description for an Active Directory forest object.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnableDiscovery<Boolean>

Specifies whether to discover Active Directory sites and subnets. You must configure an Active Directory Forest Discovery method before you use this parameter. Valid values are: \$True or \$False. The default value is \$False.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ForestFqdn<String>

Specifies the FQDN of a Configuration Manager object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Id<String[]>

Specifies an array of IDs of Configuration Manager objects. You can find the identifier value in the ForestID property of an Active Directory forest.

Aliases	ForestId
Required?	true

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies an Active Directory forest object in Configuration Manager.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PublishingPath<String>

Specifies one or more Configuration Manager sites that publish site information to an Active Directory forest. You can use a comma-separated list in quotation marks to specify more than one site.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Change the description of an Active Directory forest

This command changes the description of an Active Directory forest that has the ID 16777217 to AD Forest 01.

```
PS C:\> Set-CMActiveDirectoryForest -Id "16777217" -Description "AD Forest 01"
```

Related topics

[New-CMActiveDirectoryForest](#)

[Get-CMActiveDirectoryForest](#)

[Remove-CMActiveDirectoryForest](#)

[Get-CMActiveDirectorySite](#)

Set-CMAAlert

Set-CMAAlert

Changes properties of Configuration Manager alerts.

Syntax

Parameter Set: SetById

```
Set-CMAAlert -Id <String> [-Comments <String> ] [-NewName <String> ] [-ParameterValues <String> ] [-Severity {Error | Informational | Warning} ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetByName

```
Set-CMAAlert -Name <String> [-Comments <String> ] [-NewName <String> ] [-ParameterValues <String> ] [-Severity {Error | Informational | Warning} ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetValue

```
Set-CMAAlert -InputObject <IResultObject> [-Comments <String> ] [-NewName <String> ] [-ParameterValues <String> ] [-Severity {Error | Informational | Warning} ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Set-CMAAlert** cmdlet updates the properties of one or more Microsoft System Center 2012 Configuration Manager alerts.

Parameters

-Comments<String>

Specifies comments for an alert. These comments appear together with the alert in the Configuration Manager console.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Id<String>

Specifies an alert identifier. You can obtain the identifier of an alert by using the **Get-Alert** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a **CMAAlert** object. To obtain a **CMAAlert** object, use the **Get-CMAAlert** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies an alert name. You can obtain the name of an alert by using **Get-CMAAlert**.

Aliases	none
Required?	true
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-NewName<String>

Specifies a new name for the alert.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ParameterValues<String>

Specifies the values of administrator-defined parameters, such as thresholds.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Severity<Severities>

Specifies the severity of an alert. Valid values are:

- 1: Error
- 2: Warning
- 3: Informational

The acceptable values for this parameter are:

Error	
Informational	
Warning	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Set alert properties

This command changes the values of the *Comments* and *Severity* properties for an alert that has the ID 16777223.

```
PS C:\> Set-CMAAlert -Id "16777223" -Comments "Editing severity" -Severity 2
```

Example 2: Set alert properties by using alert object variable

In this example, the first command gets an alert object that has the ID 16777221 and stores it in the `$AlertObj` variable.

The second command changes the *Comments* property of the alert stored in the `$AlertObj` variable.

```
PS C:\> $AlertObj = Get-CMAAlert -Id "16777221"
```

```
PS C:\> Set-CMAAlert -InputObject $AlertObj -Comments "Updating alert"
```

Related topics

[Enable-CMAAlert](#)

[Get-CMAAlert](#)

[Remove-CMAAlert](#)

[Suspend-CMAAlert](#)

[Disable-CMAAlert](#)

Set-CMAAlertSubscription

Set-CMAAlertSubscription

Changes the properties of an alert subscription.

Syntax

Parameter Set: SetById

```
Set-CMAAlertSubscription -Id <String> [-AlertId <Int32[]> ] [-EmailAddress <String[]> ] [-LocaleId <Int32> ] [-NewName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetByName

```
Set-CMAAlertSubscription -Name <String> [-AlertId <Int32[]> ] [-EmailAddress <String[]> ] [-LocaleId <Int32> ] [-NewName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetValueMandatory

```
Set-CMAAlertSubscription -InputObject <IResultObject> [-AlertId <Int32[]> ] [-EmailAddress <String[]> ] [-LocaleId <Int32> ] [-NewName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetSecurityScopeById

```
Set-CMAAlertSubscription -Id <String> -SecurityScopeAction {AddMembership | RemoveMembership} -SecurityScopeName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetSecurityScopeByName

```
Set-CMAAlertSubscription -Name <String> -SecurityScopeAction {AddMembership | RemoveMembership} -SecurityScopeName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetSecurityScopeByValue

```
Set-CMAAlertSubscription -InputObject <IResultObject> -SecurityScopeAction {AddMembership | RemoveMembership} -SecurityScopeName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Set-CMAAlertSubscription** cmdlet changes the properties of an alert subscription object in Microsoft System Center 2012 Configuration Manager. You can change the name of an alert subscription, the email address of the recipient of an alert notification, the Windows locale ID, and the alert ID. You can also change the security scope membership of an alert subscription by adding it to or removing it from a specified security scope.

Parameters

-AlertId<Int32[]>

Specifies an array of alert identifiers for the subscription.

Aliases	AlertIds
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EmailAddress<String[]>

Specifies an email address where you want to send an alert notification. For example, david.chew@contoso.com. You can separate multiple email addresses by using a semicolon.

Aliases	EmailAddresses
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Id<String>

Specifies the identifier for a subscription object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-InputObject<IResultObject>

Specifies an alert notification object in Configuration Manager.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-LocaleId<Int32>

Specifies a locale for alert messages. For more information and a list of locale identifiers, see the [Locale IDs Assigned by Microsoft](http://go.microsoft.com/fwlink/?LinkId=262651) topic at <http://go.microsoft.com/fwlink/?LinkId=262651>.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies the name of an alert subscription object.

Aliases	none
Required?	true
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-NewName<String>

Specifies a new name for an alert subscription object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SecurityScopeAction<SecurityScopeActionType>

Specifies an action for a security scope. Valid values are: AddMembership and RemoveMembership.

The acceptable values for this parameter are:

AddMembership	
RemoveMembership	

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SecurityScopeName<String>

Specifies the name of a security scope. A security scope name can be Default or the name of a custom security scope.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Change the properties of an alert subscription by subscription ID

This command changes the name, email address, Windows locale ID, and alert ID of an alert subscription that has the ID 16777217.

```
PS C:\> Set-CMAAlertSubscription -Id "16777217" -NewName "Subscription02" -EmailAddress "evan.narvaez@contoso.com" -LocaleId 2057 -AlertIds 16777240
```

Example 2: Change the properties of an alert subscription by subscription name

This command changes the name, email address, Windows locale ID, and alert ID of an alert subscription named Subscription01.

```
PS C:\> Set-CMAAlertSubscription -Name "Subscription01" -NewName "Subscription02" -EmailAddress "elisa.daugherty@contoso.com" -LocaleId 2057 -AlertIds 16777240
```

Example 3: Change the properties of an alert subscription by using the output from another cmdlet as input

The first command gets an alert subscription object that has the ID 16777310 and stores the object in the \$SubObj variable.

The second command changes the properties of the alert subscription object, which include the subscription name, email recipient, locale ID, and alert ID, for the alert notification stored in the \$SubObj variable.

```
PS C:\> $SubObj = Get-CMAAlertSubscription -Id "16777310"  
PS C:\> Set-CMAAlertSubscription -AlertSubscription $SubObj -NewName "Subscription02" -EmailAddress "patti.fuller@contoso.com" -LocaleId 3081 -AlertIds 16777240
```

Example 4: Add an alert subscription to a security scope

This command adds the alert subscription named Subscription01 to the security scope named Test.

```
PS C:\> Set-CMAAlertSubscription -SecurityScopeAction AddMembership -SecurityScopeName "Test" -Name "Subscription01"
```

Example 5: Remove an alert subscription from a security scope

This command removes the alert subscription named Subscription01 from the security scope named Test.

```
PS C:\> Set-CMAAlertSubscription -SecurityScopeAction RemoveMembership -SecurityScopeName "Test" -Name "Subscription01"
```

Related topics

[New-CMAAlertSubscription](#)

[Get-CMAAlertSubscription](#)

[Remove-CMAAlertSubscription](#)

[Set-CMSecurityScope](#)

Set-CMAntiMalwarePolicy

Set-CMAntiMalwarePolicy

Changes configuration settings for an antimalware policy for Endpoint Protection.

Syntax

Parameter Set: SetByName

```
Set-CMAntiMalwarePolicy -Name <String> [-Description <String> ] [-NewName <String> ] [-Priority {Decrease | Increase} ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetAdvancedSettingsByName

```
Set-CMAntiMalwarePolicy -Name <String> [-AllowUserAddExcludes <Boolean> ] [-AllowUserConfigQuarantinedFileDeletionPeriod <Boolean> ] [-AllowUserViewHistory <Boolean> ] [-CreateSystemRestorePointBeforeClean <Boolean> ] [-DeleteQuarantinedFilesPeriod <Int32> ] [-DisableClientUI <Boolean> ] [-EnableReparsePointScanning <Boolean> ] [-RandomizeScheduledScanStartTime <Boolean> ] [-ShowNotificationMessages <Boolean> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetDefaultActionSettingsByName

```
Set-CMAntiMalwarePolicy -Name <String> [-DefaultActionHigh {Quarantine | Recommended | Remove} ] [-DefaultActionLow {Allow | None | Quarantine | Remove} ] [-DefaultActionMedium {Allow | None | Quarantine | Remove} ] [-DefaultActionSevere {Quarantine | Recommended | Remove} ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetDefintionUpdatesSettingsByName

```
Set-CMAntiMalwarePolicy -Name <String> [-AuGracePeriod <Int32> ] [-DefinitionUpdateFileSharesSources <String[]> ] [-EnableSignatureUpdateCatchUpInterval <Boolean> ] [-FallbackOrder {UpdatesDistributedFromConfigurationManager | UpdatesDistributedFromMicrosoftMalwareProtectionCenter | UpdatesDistributedFromMicrosoftUpdate | UpdatesDistributedFromWsus | UpdatesFromUncFileShares} ] [-SignatureUpdateInterval <Int32> ] [-SignatureUpdateTime <DateTime> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetExclusionSettingsByName

```
Set-CMAntiMalwarePolicy -Name <String> [-ExcludedFilePaths <String[]> ] [-ExcludedFileTypes <String[]> ] [-ExcludedProcesses <String[]> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetMicrosoftActiveProtectionServiceSettingsByName

```
Set-CMAntiMalwarePolicy -Name <String> [-AllowUserChangeSpyNetSettings <Boolean> ] [-JoinSpyNet {AdvancedMembership | BasicMembership | DoNotJoinMaps} ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetRealtimeProtectionSettingsByName

```
Set-CMAntiMalwarePolicy -Name <String> [-AllowClientUserConfigRealTime <Boolean> ] [-MonitorFileProgramActivity <Boolean> ] [-NetworkProtectionAgainstExploits <Boolean> ] [-RealTimeProtectionOn <Boolean> ] [-RealTimeScanOption {ScanIncomingAndOutgoingFiles |
```

```
ScanIncomingFilesOnly | ScanOutgoingFilesOnly} ] [-ScanAllDownloaded <Boolean> ] [-
UseBehaviorMonitor <Boolean> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetScanSettingsByName

```
Set-CMAntiMalwarePolicy -Name <String> [-AllowClientUserConfigLimitCpuUsage <Boolean> ] [-
ScanArchivedFiles <Boolean> ] [-ScanEmail <Boolean> ] [-ScanNetworkDrives <Boolean> ] [-
ScanRemovableStorage <Boolean> ] [-ScheduledScanUserControl {FullControl | NoControl |
ScanTimeOnly} ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetScheduledScanSettingsByName

```
Set-CMAntiMalwarePolicy -Name <String> [-CheckLatestDefinition <Boolean> ] [-
EnableCatchUpScan <Boolean> ] [-EnableQuickDailyScan <Boolean> ] [-EnableScheduledScan
<Boolean> ] [-LimitCpuUsage <Int32> ] [-ScanWhenClientNotInUse <Boolean> ] [-
ScheduledScanQuickTime <DateTime> ] [-ScheduledScanTime <DateTime> ] [-ScheduledScanType
{FullScan | None | QuickScan} ] [-ScheduledScanWeekday {Daily | Friday | Monday | Saturday |
Sunday | Thursday | Tuesday | Wednesday} ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetThreatOverridesSettingsByName

```
Set-CMAntiMalwarePolicy -Name <String> -OverrideAction {Allow | None | Quarantine | Remove}
-ThreatName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Set-CMAntiMalwarePolicy** cmdlet changes configuration settings for an antimalware policy for System Center 2012 Endpoint Protection. You can increase or decrease the priority by which an antimalware policy is applied, and you can apply an action to the security scope of an antimalware policy.

Parameters

-AllowClientUserConfigLimitCpuUsage<Boolean>

Indicates whether users on client computers are allowed to limit CPU usage.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AllowClientUserConfigRealTime<Boolean>

Indicates whether users on client computers are allowed to configure real-time protection settings.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AllowUserAddExcludes<Boolean>

Indicates whether users are allowed to exclude files and folders, file types, and processes.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AllowUserChangeSpyNetSettings<Boolean>

Indicates whether users are allowed to modify Microsoft Active Protection Service settings.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AllowUserConfigQuarantinedFileDeletionPeriod<Boolean>

Indicates whether users are allowed to configure the setting for quarantined file deletion.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AllowUserViewHistory<Boolean>

Indicates whether users are allowed to view the full History results.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AuGracePeriod<Int32>

Specifies the number of hours after which clients update from alternative sources. If Configuration Manager is used as a source for definition updates, clients only update from alternative sources if the definition is older than the specified amount of hours.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CheckLatestDefinition<Boolean>

Indicates whether the policy checks for the latest definition updates before running a scan.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CreateSystemRestorePointBeforeClean<Boolean>

Indicates whether the cmdlet creates a system restore point before computers are cleaned.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DefaultActionHigh<DefaultActionSevereAndHighType>

Specifies the default action taken for the High alert level. Valid values are:

- Quarantine
- Recommended
- Remove

The acceptable values for this parameter are:

Quarantine	
Recommended	
Remove	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DefaultActionLow<DefaultActionMediumAndLowType>

Specifies the default action taken for the Low alert level. Valid values are:

- Allow
- None
- Quarantine
- Remove

The acceptable values for this parameter are:

Allow	
None	
Quarantine	
Remove	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DefaultActionMedium<DefaultActionMediumAndLowType>

Specifies the default action taken for the Medium alert level. Valid values are:

- Allow
- None
- Quarantine
- Remove

The acceptable values for this parameter are:

Allow	
None	
Quarantine	
Remove	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DefaultActionSevere<DefaultActionSevereAndHighType>

Specifies the default action taken for the Severe alert level. Valid values are:

- Quarantine
- Recommended
- Remove

The acceptable values for this parameter are:

Quarantine	
Recommended	
Remove	

Aliases	none
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DefinitionUpdateFileSharesSources<String[]>

Specifies the sources and order for Endpoint Protection definition updated.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeleteQuarantinedFilesPeriod<Int32>

Specifies the number of days after which quarantined files are deleted.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Description<String>

Specifies a description for the antimalware policy.

Aliases	none
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DisableClientUI<Boolean>

Indicates whether the client user interface is disabled.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnableCatchUpScan<Boolean>

Indicates whether a scan of the selected scan type is forced if a client computer is offline during two or more scheduled scans.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnableQuickDailyScan<Boolean>

Indicates that a daily quick scan is run on client computers.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnableReparsePointScanning<Boolean>

Indicates whether reparse point scanning is enabled.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnableScheduledScan<Boolean>

Indicates whether a scheduled scan is run on client computers.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnableSignatureUpdateCatchUpInterval<Boolean>

Indicates whether the policy forces a definition update if the client computer is offline for more than two consecutive scheduled updates.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ExcludedFilePaths<String[]>

Specifies an array of excluded files and folders.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ExcludedFileTypes<String[]>

Specifies an array of excluded file types.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ExcludedProcesses<String[]>

Specifies an array of excluded processes.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-FallbackOrder<FallbackOrderType[]>

Specifies an array of fallback order types. Valid values are:

- UpdatesDistributedFromConfigurationManager
- UpdatesDistributedFromMicrosoftMalwareProtectionCenter
- UpdatesDistributedFromMicrosoftUpdate
- UpdatesDistributedFromWsus
- UpdatesFromUncFileShares

The acceptable values for this parameter are:

UpdatesDistributedFromConfigurationManager	
UpdatesDistributedFromMicrosoftMalwareProtectionCenter	
UpdatesDistributedFromMicrosoftUpdate	
UpdatesDistributedFromWsus	
UpdatesFromUncFileShares	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-JoinSpyNet<JoinSpyNetType>

Specifies the Microsoft Active Protection Service membership type. Valid values are:

-- AdvancedMembership

-- BasicMembership

-- DoNotJoinMaps

The acceptable values for this parameter are:

AdvancedMembership	
BasicMembership	
DoNotJoinMaps	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-LimitCpuUsage<Int32>

Specifies the limit CPU usage during scans, in percentage.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-MonitorFileProgramActivity<Boolean>

Indicates whether file and program activity is monitored on the computer.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies the name of an antimalware policy.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-NetworkProtectionAgainstExploits<Boolean>

Indicates whether protection against network-based exploits is enabled.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-NewName<String>

Specifies a new name for the antimalware policy.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OverrideAction<DefaultActionMediumAndLowType>

Specifies the threat override action. Use this parameter with the *ThreatName* parameter to configure threat override settings. Valid values are:

- Allow
- None
- Quarantine
- Remove

The acceptable values for this parameter are:

Allow	
None	
Quarantine	
Remove	

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Priority<PriorityChangeType>

Specifies the priority of an antimalware policy. Valid values are:

- Increase
- Decrease

The acceptable values for this parameter are:

Decrease	
Increase	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RandomizeScheduledScanStartTime<Boolean>

Indicates whether scheduled scan and definition update start times are randomized (within 30 minutes).

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RealTimeProtectionOn<Boolean>

Indicates whether real-time protection is enabled.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-RealTimeScanOption<RealTimeScanOptionType>

Specifies the system files scan type. Valid values are:

- ScanIncomingAndOutgoingFiles
- ScanIncomingFilesOnly
- ScanOutgoingFilesOnly

The acceptable values for this parameter are:

ScanIncomingAndOutgoingFiles	
ScanIncomingFilesOnly	
ScanOutgoingFilesOnly	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ScanAllDownloaded<Boolean>

Indicates whether all downloaded files and attachments are scanned.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ScanArchivedFiles<Boolean>

Indicates whether archived files are scanned.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ScanEmail<Boolean>

Indicates whether email and email attachments are scanned.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ScanNetworkDrives<Boolean>

Indicates whether network drives are scanned when running a full scan.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ScanRemovableStorage<Boolean>

Indicates whether removable storage devices, such as USB drives, are scanned.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ScanWhenClientNotInUse<Boolean>

Indicates whether a scheduled scan is started only when the computer is idle.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ScheduledScanQuickTime<DateTime>

Specifies the date and time that a daily quick scan is scheduled. To obtain a **DateTime** object, use the **Get-Date** cmdlet. For more information, type `Get-Help Get-Date`.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ScheduledScanTime<DateTime>

Specifies the time of a scheduled scan. To obtain a **DateTime** object, use the **Get-Date** cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ScheduledScanType<ScheduledScanType>

Specifies the type of a scheduled scan. Valid values are:

- FullScan
- None
- QuickScan

The acceptable values for this parameter are:

FullScan	
None	
QuickScan	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ScheduledScanUserControl<ScheduledScanUserControlType>

Specifies the user control of scheduled scans. Valid values are:

- FullControl

- NoControl
- ScanTimeOnly

The acceptable values for this parameter are:

FullControl	
NoControl	
ScanTimeOnly	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ScheduledScanWeekday<ScheduledScanWeekdayType>

Specifies the day of the week a scheduled scan runs. Valid values are:

- Daily
- Monday
- Tuesday
- Wednesday
- Thursday
- Friday
- Saturday
- Sunday

The acceptable values for this parameter are:

Daily	
Friday	
Monday	
Saturday	
Sunday	

Thursday	
Tuesday	
Wednesday	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ShowNotificationMessages<Boolean>

Indicates whether notification messages are shown on the client computer when the user needs to run a full scan, update definitions, or run Windows Defender Offline.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SignatureUpdateInterval<Int32>

Specifies the interval, in hours, that the policy checks for Endpoint Protection definitions. Specify 0 to disable the check on interval.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SignatureUpdateTime<DateTime>

Specifies the time that the policy checks for Endpoint Protection definitions. To obtain a **DateTime** object, use the **Get-Date** cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ThreatName<String>

Specifies the name of a threat. Use this parameter with the *OverrideAction* parameter to configure threat override settings.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UseBehaviorMonitor<Boolean>

Indicates whether behavior monitoring is enabled.

Aliases	none
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Increase the priority of an antimalware policy

This command increases the priority of the antimalware policy named ContosoPolicy.

PS C:\> Set-CMAntiMalwarePolicy -Priority Increase -Name "ContosoPolicy"

Related topics

[Export-CMAntimalwarePolicy](#)

[Get-CMAntiMalwarePolicy](#)

[Import-CMAntimalwarePolicy](#)

[Merge-CMAntimalwarePolicy](#)

[New-CMAntimalwarePolicy](#)

[Remove-CMAntiMalwarePolicy](#)

Set-CMApplication

Set-CMApplication

Sets properties of an application in Configuration Manager.

Syntax

Parameter Set: SetById

```
Set-CMApplication -Id <String[]> [-AppCategories <String[]> ] [-AutoInstall <Boolean> ] [-Description <String> ] [-DistributionPointSetting {AutoDownload | DeltaCopy | NoDownload} ] [-DistributionPriority {High | Low | Medium} ] [-IconLocationFile <String> ] [-IsFeatured <Boolean> ] [-Keyword <String> ] [-LinkText <String> ] [-LocalizedApplicationDescription <String> ] [-LocalizedApplicationName <String> ] [-NewName <String> ] [-OptionalReference <String> ] [-Owner <String> ] [-PrivacyUrl <String> ] [-Publisher <String> ] [-ReleaseDate <DateTime> ] [-SendToProtectedDistributionPoint <Boolean> ] [-SoftwareVersion <String> ] [-SupportContact <String> ] [-UserCategories <String[]> ] [-UserDocumentation <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetByName

```
Set-CMApplication -Name <String[]> [-AppCategories <String[]> ] [-AutoInstall <Boolean> ] [-Description <String> ] [-DistributionPointSetting {AutoDownload | DeltaCopy | NoDownload} ] [-DistributionPriority {High | Low | Medium} ] [-IconLocationFile <String> ] [-IsFeatured <Boolean> ] [-Keyword <String> ] [-LinkText <String> ] [-LocalizedApplicationDescription <String> ] [-LocalizedApplicationName <String> ] [-NewName <String> ] [-OptionalReference <String> ] [-Owner <String> ] [-PrivacyUrl <String> ] [-Publisher <String> ] [-ReleaseDate <DateTime> ] [-SendToProtectedDistributionPoint <Boolean> ] [-SoftwareVersion <String> ] [-SupportContact <String> ] [-UserCategories <String[]> ] [-UserDocumentation <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetValue

```
Set-CMApplication -InputObject <IResultObject> [-AppCategories <String[]> ] [-AutoInstall <Boolean> ] [-Description <String> ] [-DistributionPointSetting {AutoDownload | DeltaCopy | NoDownload} ] [-DistributionPriority {High | Low | Medium} ] [-IconLocationFile <String> ] [-IsFeatured <Boolean> ] [-Keyword <String> ] [-LinkText <String> ] [-LocalizedApplicationDescription <String> ] [-LocalizedApplicationName <String> ] [-NewName <String> ] [-OptionalReference <String> ] [-Owner <String> ] [-PrivacyUrl <String> ] [-Publisher <String> ] [-ReleaseDate <DateTime> ] [-SendToProtectedDistributionPoint <Boolean> ] [-SoftwareVersion <String> ] [-SupportContact <String> ] [-UserCategories <String[]> ] [-UserDocumentation <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetSecurityScopeById

```
Set-CMApplication -Id <String[]> -SecurityScopeAction {AddMembership | RemoveMembership} -SecurityScopeName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetSecurityScopeByName

```
Set-CMApplication -Name <String[]> -SecurityScopeAction {AddMembership | RemoveMembership} -SecurityScopeName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetSecurityScopeByValue

```
Set-CMApplication -InputObject <IResultObject> -SecurityScopeAction {AddMembership | RemoveMembership} -SecurityScopeName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Set-CMApplication** cmdlet sets properties of an application.

Parameters

-AppCategories<String[]>

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AutoInstall<Boolean>

Specifies whether the task sequence action can install the application.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Description<String>

Specifies a description for the application.

Aliases	LocalizedDescription
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DistributionPointSetting<DistributionPointSettingType>

Specifies the pre-staged distribution point settings. Valid values are:

- AutoDownload. Automatically download content when packages are assigned to distribution points.
- DeltaCopy. Download only content changes to distribution points.
- NoDownload. Manually copy the content in this package to distribution points.

The acceptable values for this parameter are:

AutoDownload	
DeltaCopy	
NoDownload	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DistributionPriority<DistributionPriorityType>

Specifies the order in which packages are sent to other sites. Packages with a high priority are sent before packages with a low or medium priority. Packages with equal priority are sent in the order they are created.

The acceptable values for this parameter are:

High	
Low	
Medium	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-IconLocationFile<String>

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Id<String[]>

Specifies an array of application IDs.

Aliases	CId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-InputObject<IResultObject>

Specifies an application object. To obtain an application object, use the **Get-CMApplication** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-IsFeatured<Boolean>

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Keyword<String>

Specifies a key word for the application.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-LinkText<String>

Specifies a description that appears in the Application Catalog. The text accompanies the hyperlink to additional information or documentation about this application.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-LocalizedApplicationDescription<String>

Specifies the localized description string that appears in the client software center or catalog web site.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-LocalizedApplicationName<String>

Specifies the localized name string that appears in the client software center or catalog web site.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String[]>

Specifies an array of names for the application.

Aliases	LocalizedDisplayName
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-NewName<String>

Specifies a new name for the application.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OptionalReference<String>

Specifies the optional reference information for this application.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Owner<String>

Specifies one or more administrative users who are the owners of this application.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PrivacyUrl<String>

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Publisher<String>

Specifies the name of a software publisher in Configuration Manager.

Aliases	Manufacturer
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ReleaseDate<DateTime>

Specifies a release date of the application.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SecurityScopeAction<SecurityScopeActionType>

Specifies an action for a security scope. Valid values are: AddMembership and RemoveMembership.

The acceptable values for this parameter are:

AddMembership	
RemoveMembership	

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SecurityScopeName<String>

Specifies the name of a security scope. A security scope name can be Default or the name of a custom security scope.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SendToProtectedDistributionPoint<Boolean>

Specifies whether to copy this application to protected distribution points.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SoftwareVersion<String>

Specifies a software version for an application.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SupportContact<String>

Specifies one or more administrative users who are support contacts for this application.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UserCategories<String[]>

Specifies an array of categories for which software is a member. Use this parameter to identify a group, or category, of software, such as office productivity or graphics.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UserDocumentation<String>

Specifies a hyperlink, in Uniform Resource Indicator (URI) format, to additional information for this application.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Set an application

This example sets security scope action for an application.

```
PS C:\> Set-CMApplication -SecurityScopeAction AddMembership -SecurityScopeName "22" -Name "adw"
```

Related topics

[Export-CMApplication](#)

[Get-CMApplication](#)

[Import-CMApplication](#)

[New-CMApplication](#)

[Remove-CMApplication](#)

[Resume-CMApplication](#)

[Suspend-CMApplication](#)

Set-CMApplicationCatalogWebsitePoint

Set-CMApplicationCatalogWebsitePoint

Changes the settings of an Application Catalog website point.

Syntax

Parameter Set: SetByName

```
Set-CMApplicationCatalogWebsitePoint -SiteCode <String> -SiteSystemServerName <String> [-ClientCommunicationType {HTTP | HTTPS} ] [-ClientConnectionType {Internet | InternetAndIntranet | Intranet} ] [-ColorBlue <Int32> ] [-ColorGreen <Int32> ] [-ColorRed <Int32> ] [-NetbiosName <String> ] [-OrganizationName <String> ] [-SiteSystemServerNameConfiguredForApplicationCatalogWebServicePoint <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetByValue

```
Set-CMApplicationCatalogWebsitePoint -InputObject <IResultObject> [-ClientCommunicationType {HTTP | HTTPS} ] [-ClientConnectionType {Internet | InternetAndIntranet | Intranet} ] [-ColorBlue <Int32> ] [-ColorGreen <Int32> ] [-ColorRed <Int32> ] [-NetbiosName <String> ] [-OrganizationName <String> ] [-SiteSystemServerNameConfiguredForApplicationCatalogWebServicePoint <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Set-CMApplicationCatalogWebsitePoint** cmdlet changes the settings of an Application Catalog website point in Microsoft System Center 2012 Configuration Manager. This site system role supports the Application Catalog website.

Specify the site that this website point supports and the server that hosts the website point. You can specify the website name and NetBIOS name of the Application Catalog.

You can customize the page that users see when they connect to the Application Catalog. Specify custom values for the colors blue, green, and red. You can also specify a name for users to see in the browser, such as a company name or a division within a company.

Parameters

-ClientCommunicationType<ComputerCommunicationType>

Specifies the client communication type. Valid values are:

-- HTTP

-- HTTPS

Specify HTTPS to connect by using the more secure setting and to determine whether clients connect from the Internet. This option requires a PKI certificate on the server for server authentication to clients and for encryption of data over Secure Socket Layer (SSL).

The acceptable values for this parameter are:

HTTP	
HTTPS	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ClientConnectionType<ClientConnectionTypes>

Specifies the client connection type. Valid values are:

- Internet
- InternetAndIntranet
- Intranet

The acceptable values for this parameter are:

Internet	
InternetAndIntranet	
Intranet	

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ColorBlue<Int32>

Specifies an integer value for a custom blue color. Configuration Manager uses custom colors to conform to customer branding.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ColorGreen<Int32>

Specifies an integer value for a custom green color. Configuration Manager uses custom colors to conform to customer branding.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ColorRed<Int32>

Specifies an integer value for a custom red color. Configuration Manager uses custom colors to conform to customer branding.

Aliases	none
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies an Application Catalog website point object. To obtain Application Catalog website point object, use the **Get-CMApplicationCatalogWebsitePoint** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-NetbiosName<String>

Specifies the NetBIOS name of the server that hosts the Application Catalog website point.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OrganizationName<String>

Specifies a name for a customer organization. This name appears to users who access the Application Catalog.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteCode<String>

Specifies the site code for a Configuration Manager site.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteSystemServerName<String>

Specifies the name of a server that hosts a site system role.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-

SiteSystemServerNameConfiguredForApplicationCatalogWebServicePoint<String>

Specifies the name of a server that hosts the site system role for the Application Catalog web service point.

The Application Catalog web service point is a site system role that provides information about available software from the Software Library to the Application Catalog website.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Change the settings of an Application Catalog website point

This command changes the settings of the Application Catalog website point hosted on the site system server named CMDIV-WEST04.CORP.CONTOSO.COM. The command specifies that the NetBIOS server named CMDIV-WEST02 hosts the Application Catalog website point. The command specifies that Configuration Manager displays the organization name MarketingWest in the Software Center, and sets the custom colors for the Software Center.

```
PS C:\> Set-CMApplicationCatalogWebsitePoint -SiteSystemServerName "CMDIV-  
WEST04.CORP.CONTOSO.COM" -SiteCode "CM4" -NetbiosName "CMDIV-WEST02" -OrganizationName  
"MarketingWest" -ColorRed 168 -ColorGreen 201 -ColorBlue 52
```

Example 2: Change the settings of an Application Catalog website point by using an object variable

The first command uses the **Get-CMApplicationCatalogWebsitePoint** cmdlet to get the Application Catalog website point that is associated with the site system server named CMDIV-WEST04.CORP.CONTOSO.COM that is hosted on the Configuration Manager site that has the site code CM4. The command stores the result in the \$Acwp variable.

The second command changes the settings of the Application Catalog website point stored in \$Acwp. The command specifies that the NetBIOS server named CMDIV-WEST02 hosts the Application Catalog website point. The command specifies that Configuration Manager displays the organization name MarketingWest in the Software Center, and sets the custom colors for the Software Center.

```
PS C:\> $Acwp= Get-CMApplicationCatalogWebsitePoint -SiteSystemServerName "CMDIV-  
WEST04.CORP.CONTOSO.COM" -SiteCode "CM4"  
PS C:\> Set-CMApplicationCatalogWebsitePoint -InputObject $Acwp -NetbiosName "CMDIV-WEST02"  
-OrganizationName "MarketingWest" -ColorR 160 -ColorG 200 -ColorB 50
```

Related topics

[Add-CMApplicationCatalogWebsitePoint](#)

[Get-CMApplicationCatalogWebsitePoint](#)

[Remove-CMApplicationCatalogWebSitePoint](#)

[Remove-CMApplicationCatalogWebSitePoint](#)

[Get-CMApplicationCatalogWebServicePoint](#)

Set-CMApplicationDeployment

Set-CMApplicationDeployment

Modifies properties for an application deployment in Configuration Manager.

Syntax

Parameter Set: SetApplicationDeploymentByIdMandatory

```
Set-CMApplicationDeployment -ApplicationId <String> -CollectionName <String> [-AppRequiresApproval <Boolean> ] [-AvaliableDate <DateTime> ] [-AvaliableTime <DateTime> ] [-Comment <String> ] [-CreateAlertBaseOnPercentFailure <Boolean> ] [-CreateAlertBaseOnPercentSuccess <Boolean> ] [-DeadlineDate <DateTime> ] [-DeadlineTime <DateTime> ] [-EnableMomAlert <Boolean> ] [-FailParameterValue <Int32> ] [-OverrideServiceWindow <Boolean> ] [-PersistOnWriteFilterDevice <Boolean> ] [-PostponeDate <DateTime> ] [-PostponeTime <DateTime> ] [-PreDeploy <Boolean> ] [-RaiseMomAlertsOnFailure <Boolean> ] [-RebootOutsideServiceWindow <Boolean> ] [-SendWakeUpPacket <Boolean> ] [-SuccessParameterValue <Int32> ] [-TimeBaseOn {LocalTime | UTC} ] [-UseMeteredNetwork <Boolean> ] [-UserNotification {DisplayAll | DisplaySoftwareCenterOnly | HideAll} ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetApplicationDeploymentByNameMandatory

```
Set-CMApplicationDeployment -ApplicationName <String> -CollectionName <String> [-AppRequiresApproval <Boolean> ] [-AvaliableDate <DateTime> ] [-AvaliableTime <DateTime> ] [-Comment <String> ] [-CreateAlertBaseOnPercentFailure <Boolean> ] [-CreateAlertBaseOnPercentSuccess <Boolean> ] [-DeadlineDate <DateTime> ] [-DeadlineTime <DateTime> ] [-EnableMomAlert <Boolean> ] [-FailParameterValue <Int32> ] [-OverrideServiceWindow <Boolean> ] [-PersistOnWriteFilterDevice <Boolean> ] [-PostponeDate <DateTime> ] [-PostponeTime <DateTime> ] [-PreDeploy <Boolean> ] [-RaiseMomAlertsOnFailure <Boolean> ] [-RebootOutsideServiceWindow <Boolean> ] [-SendWakeUpPacket <Boolean> ] [-SuccessParameterValue <Int32> ] [-TimeBaseOn {LocalTime | UTC} ] [-UseMeteredNetwork <Boolean> ] [-UserNotification {DisplayAll | DisplaySoftwareCenterOnly | HideAll} ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetApplicationDeploymentByValueMandatory

```
Set-CMApplicationDeployment -Application <IResultObject> -CollectionName <String> [-AppRequiresApproval <Boolean> ] [-AvaliableDate <DateTime> ] [-AvaliableTime <DateTime> ] [-Comment <String> ] [-CreateAlertBaseOnPercentFailure <Boolean> ] [-CreateAlertBaseOnPercentSuccess <Boolean> ] [-DeadlineDate <DateTime> ] [-DeadlineTime <DateTime> ] [-EnableMomAlert <Boolean> ] [-FailParameterValue <Int32> ] [-OverrideServiceWindow <Boolean> ] [-PersistOnWriteFilterDevice <Boolean> ] [-PostponeDate <DateTime> ] [-PostponeTime <DateTime> ] [-PreDeploy <Boolean> ] [-RaiseMomAlertsOnFailure <Boolean> ] [-RebootOutsideServiceWindow <Boolean> ] [-SendWakeUpPacket <Boolean> ] [-SuccessParameterValue <Int32> ] [-TimeBaseOn {LocalTime | UTC} ] [-UseMeteredNetwork <Boolean> ] [-UserNotification {DisplayAll | DisplaySoftwareCenterOnly | HideAll} ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Set-CMApplicationDeployment** cmdlet modifies properties for an application deployment in Microsoft System Center 2012 Configuration Manager. An application deployment installs an application according to schedule for a several computers. Application deployments can also allow users to install at a time they choose.

To specify an application deployment to modify, specify the collection name and the application. You can specify an application by name or ID, or you can use the **Get-CMApplication** cmdlet to get an application to modify.

Parameters

-Application<IResultObject>

Specifies an application object. To obtain an application object, use the **Get-CMApplication** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ApplicationId<String>

Specifies the ID of an application.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ApplicationName<String>

Specifies the name of an application.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AppRequiresApproval<Boolean>

Indicates whether this application requires Administrator approval if the application is available to the user.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AvailableDate<DateTime>

Specifies a date as a **DateTime** object. To obtain a **DateTime** object, use the **Get-Date** cmdlet. For more information, type `Get-Help Get-Date`. This is the date on which the deployment becomes available. If you specify a value for the *DeployAvailableTime* parameter in addition to this parameter, the cmdlet uses that value.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-AvailableTime<DateTime>

Specifies a date as a **DateTime** object. To obtain a **DateTime** object, use the **Get-Date** cmdlet. This is the date on which the deployment becomes available.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CollectionName<String>

Specifies the name of device collection or user collection.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Comment<String>

Specifies a comment for the deployment.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CreateAlertBaseOnPercentFailure<Boolean>

Indicates whether to create an alert for a percentage of the applications that fail to deploy. Enter the percentage value by using the *FailParameterValue* parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CreateAlertBaseOnPercentSuccess<Boolean>

Indicates whether to create an alert for a percentage of the applications that deploy successfully. Enter the percentage value by using the *SuccessParameterValue* parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeadlineDate<DateTime>

Specifies a date as a **DateTime** object. To obtain a **DateTime** object, use the **Get-Date** cmdlet. This is the deadline for deployment of the application. If the application has not deployed to a client by this date, Configuration Manager forces deployment. If you specify a value for the *DeadlineTime* parameter in addition to this parameter, the cmdlet uses that value.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeadlineTime<DateTime>

Specifies a time as a **DateTime** object. To obtain a **DateTime** object, use the **Get-Date** cmdlet. This is the deadline for deployment of the application. If the application has not deployed to a client by this time, Configuration Manager forces deployment.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnableMomAlert<Boolean>

Indicates whether alerts from this cmdlet appear in System Center 2012 – Operations Manager.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-FailParameterValue<Int32>

Specifies the percentage of failed application installation that causes an alert. Specify an integer from 1 through 100. You must also specify the *CreatAlertBaseOnPercentFailure* parameter as \$True.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OverrideServiceWindow<Boolean>

Indicates whether the deployment takes place even if scheduled outside of a service window. A service window is a specified period of time used for computer maintenance and updates. If this value is \$True, Configuration Manager deploys the application even the scheduled time falls outside the service window. If this value is \$False, Configuration Manager does not deploy the application outside the service window, but Configuration Manager waits until it can deploy in a service window.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PersistOnWriteFilterDevice<Boolean>

Indicates whether to enable write filters for embedded devices. For a value of \$True, the device commits changes during a maintenance window. This action requires a restart. For a value of \$False, the device saves changes in an overlay and commits them later.

Aliases	none
Required?	false
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PostponeDate<DateTime>

Specifies a date as a **DateTime** object. To obtain a **DateTime** object, use the **Get-Date** cmdlet. Configuration Manager generates a deployment alert after this length of time. If you specify a value for the *PostponeTime* parameter in addition to this parameter, the cmdlet uses that value.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PostponeTime<DateTime>

Specifies a time as a **DateTime** object. To obtain a **DateTime** object, use the **Get-Date** cmdlet. Configuration Manager generates a deployment alert after this length of time.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PreDeploy<Boolean>

Indicates whether to pre-deploy the application to the primary device of the user.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RaiseMomAlertsOnFailure<Boolean>

Indicates whether to create an Operations Manager alert if a client fails to install the application.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RebootOutsideServiceWindow<Boolean>

Indicates whether a computer restarts outside a service window. A service window is a specified period of time used for computer maintenance and updates. If this value is \$True, any required restart takes place without regard to service windows. If this value is \$False, the computer does not restart outside a service window.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SendWakeUpPacket<Boolean>

Indicates whether to send a wake up packet to computers before the deployment begins. If this value is \$True, Configuration Manager wakes a computer from sleep. If this value is \$False, it does not wake computers from sleep. For computers to wake, you must first configure Wake On LAN.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SuccessParameterValue<Int32>

Specifies the percentage of successful application installation that causes an alert. Specify an integer from 0 through 99. You must also specify the *CreatAlertBaseOnPercentSuccess* parameter as \$True.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-TimeBaseOn<TimeType>

Specifies which time zone to use. Valid values are:

-- LocalTime. Use local time.

-- UTC. Use Coordinated Universal Time (UTC), also known as Greenwich Mean Time.

The acceptable values for this parameter are:

LocalTime	
UTC	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UseMeteredNetwork<Boolean>

Indicates whether to allow clients to download content over a metered Internet connection after the deadline, which may incur additional expense.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UserNotification<UserNotificationType>

Specifies the type of notification for a client. Valid values are:

- DisplayAll
- DisplaySoftwareCenterOnly
- HideAll

The acceptable values for this parameter are:

DisplayAll	
DisplaySoftwareCenterOnly	
HideAll	

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Modify availability and deadline for an application deployment

This command modifies an application deployment for an application named Track System 2011 for a collection named All Users. The command specifies the date and time when the application becomes available and the date and time of a deadline for deployment.

```
PS C:\> Set-CMApplicationDeployment -ApplicationName "Track System 2011" -CollectionName  
"All Users" -AvaliableDate 2012/10/21 -AvaliableTime 17:25 -DeadlineDate 2013/01/01 -  
DeadlineTime 13:10
```

Related topics

[Start-CMApplicationDeployment](#)

[Start-CMApplicationDeploymentSimulation](#)

[Get-CMApplication](#)

Set-CMAppVVirtualEnvironment

Set-CMAppVVirtualEnvironment

Changes settings for virtual applications that you have deployed by using Configuration Manager.

Syntax

Parameter Set: SetSecurityScopeById

```
Set-CMAppVVirtualEnvironment -Id <String[]> -SecurityScopeAction {AddMembership | RemoveMembership} -SecurityScopeName <String> [-SecuredScopeNames <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetById

```
Set-CMAppVVirtualEnvironment -Id <String[]> [-AddApplicationGroup <VirtualEnvironmentGroup[]> ] [-Description <String> ] [-NewName <String> ] [-RemoveApplicationGroup <VirtualEnvironmentGroup[]> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetByName

```
Set-CMAppVVirtualEnvironment -Name <String[]> [-AddApplicationGroup <VirtualEnvironmentGroup[]> ] [-Description <String> ] [-NewName <String> ] [-RemoveApplicationGroup <VirtualEnvironmentGroup[]> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetByValue

```
Set-CMAppVVirtualEnvironment -InputObject <IResultObject> [-AddApplicationGroup <VirtualEnvironmentGroup[]> ] [-Description <String> ] [-NewName <String> ] [-RemoveApplicationGroup <VirtualEnvironmentGroup[]> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetSecurityScopeByName

```
Set-CMAppVVirtualEnvironment -Name <String[]> -SecurityScopeAction {AddMembership | RemoveMembership} -SecurityScopeName <String> [-SecuredScopeNames <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetSecurityScopeByValue

```
Set-CMAppVVirtualEnvironment -InputObject <IResultObject> -SecurityScopeAction {AddMembership | RemoveMembership} -SecurityScopeName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Set-CMAppVVirtualEnvironment** cmdlet changes settings for one or more Microsoft Application Virtualization (App-V) virtual environment objects from Microsoft System Center 2012 Configuration Manager. You can specify App-V environments by name or ID.

Parameters

-AddApplicationGroup<VirtualEnvironmentGroup[]>

Specifies an array of application groups to add. Application groups contain multiple App-V deployment types that run in the same environment.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Description<String>

Specifies a description for the App-V virtual environment.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Id<String[]>

Specifies an array of IDs of virtual environments.

Aliases	CId
---------	-----

Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a virtual environment object for Configuration Manager. To obtain a virtual environment object, use the **Get-CMAppVVirtualEnvironment** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String[]>

Specifies an array of names of virtual environments.

Aliases	LocalizedDisplayName
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-NewName<String>

Specifies a new name for a virtual environment.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RemoveApplicationGroup<VirtualEnvironmentGroup[]>

Specifies an array of application groups to remove. Application groups contain multiple App-V deployment types that run in the same environment.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SecuredScopeNames<String>

Specifies the name of security scopes. A security scope can be Default or the name of a custom-created security scope.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SecurityScopeAction<SecurityScopeActionType>

Specifies an action for a security scope. Valid values are: AddMembership and RemoveMembership.

The acceptable values for this parameter are:

AddMembership	
RemoveMembership	

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SecurityScopeName<String>

Specifies the name of a security scope. A security scope name can be Default or the name of a custom security scope.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Change virtual environment settings by using a name

This command removes the virtual environment named VMWin03 from the security scope named ClientSecGroup01.

```
PS C:\> Set-CMAppVVirtualEnvironment -Name "VMWin03" -SecurityScopeAction RemoveMembership -SecurityScopeName "ClientSecGroup01"
```

Related topics

[Get-CMAppVVirtualEnvironment](#)

[New-CMAppVVirtualEnvironment](#)

[Remove-CMAppVVirtualEnvironment](#)

Set-CMAssetIntelligenceCatalogItem

Set-CMAssetIntelligenceCatalogItem

Changes the properties of an item in the Asset Intelligence catalog.

Syntax

Parameter Set: SetById

```
Set-CMAssetIntelligenceCatalogItem -Id <String[]> [-Description <String> ] [-LanguageId <Int32> ] [-NewCategoryName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetByName

```
Set-CMAssetIntelligenceCatalogItem -CategoryName <String> [-Description <String> ] [-LanguageId <Int32> ] [-NewCategoryName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetValueMandatory

```
Set-CMAssetIntelligenceCatalogItem -InputObject <IResultObject> [-Description <String> ] [-LanguageId <Int32> ] [-NewCategoryName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Set-CMAssetIntelligenceCatalogItem** cmdlet changes the properties of software categories, software families, and custom software labels in the Microsoft System Center 2012 Configuration Manager Asset Intelligence catalog.

The Asset Intelligence catalog contains categorization and identification information for software titles. The catalog includes predefined categories and families. Predefined items cannot be modified. In addition to predefined software categories and software families, you can create custom categories and families. You can also create custom software labels.

Parameters

-CategoryName<String>

Specifies the name of a category, family, or label in the Asset Intelligence catalog.

Aliases	none
Required?	true
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Description<String>

Specifies the description of a category, family, or label in the Asset Intelligence catalog.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Id<String[]>

Specifies an array of IDs of asset intelligence catalog items.

Aliases	CategoryId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies an Asset Intelligence catalog item. To obtain an Asset Intelligence catalog item, use the **Get-CAAssetIntelligenceCatalogItem** cmdlet.

Aliases	none
Required?	true

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-LanguageId<Int32>

Specifies the locale identifier for an item. For more information and a list of locale identifiers, see [Locale IDs Assigned by Microsoft](http://go.microsoft.com/fwlink/?LinkId=262651) (http://go.microsoft.com/fwlink/?LinkId=262651).

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-NewCategoryName<String>

Specifies a new category name for a category, family, or label in the Asset Intelligence catalog.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Change the properties of a catalog item by ID

This command changes the category name, description, and language ID for the Asset Intelligence catalog item that has the category ID 1211.

```
PS C:\> Set-CMAssetIntelligenceCatalogItem -Id "1211" -NewCategoryName "Windows Databases" -
Description "Windows-based databases" -LanguageId 1033
```

Example 2: Change the properties of a catalog item category of items

This command changes the category name, description, and language ID for the Asset Intelligence catalog item that has the category name Database Tools.

```
PS C:\> Set-CMAssetIntelligenceCatalogItem -CategoryName "Database Tools" -NewCategoryName
"Database Clients" -Description "Database client software" -LanguageId 1033
```

Example 3: Rename a category

This command changes the category name of the Asset Intelligence catalog item that has the category name Database Clients to Database Server Tools.

```
PS C:\> Set-CMAssetIntelligenceCatalogItem -CategoryName "Database Clients" -NewCategoryName "Database Server Tools"
```

Related topics

[New-CMAssetIntelligenceCatalogItem](#)

[Get-CMAssetIntelligenceCatalogItem](#)

[Remove-CMAssetIntelligenceCatalogItem](#)

Set-CMAssetIntelligenceClass

Set-CMAssetIntelligenceClass

Modifies the Asset Intelligence hardware inventory reporting classes.

Syntax

Parameter Set: SetByAllReportClass

```
Set-CMAssetIntelligenceClass -EnableAllReportingClass [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SetBySelectReportClass

```
Set-CMAssetIntelligenceClass [-DisableReportingClass {SMS_AutoStartSoftware |  
SMS_BrowserHelperObject | SMS_InstalledExecutable | SMS_InstalledSoftware |  
SMS_SoftwareShortcut | SMS_SoftwareTag | SMS_SystemConsoleUsage | SMS_SystemConsoleUser |  
SoftwareLicensingProduct | SoftwareLicensingService | Win32_USBDevice} ] [-  
EnableReportingClass {SMS_AutoStartSoftware | SMS_BrowserHelperObject |  
SMS_InstalledExecutable | SMS_InstalledSoftware | SMS_SoftwareShortcut | SMS_SoftwareTag |  
SMS_SystemConsoleUsage | SMS_SystemConsoleUser | SoftwareLicensingProduct |  
SoftwareLicensingService | Win32_USBDevice} ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Set-CMAssetIntelligenceClass** cmdlet modifies the Asset Intelligence hardware inventory reporting classes. The Hardware Inventory Client Agent collects inventory from Microsoft System Center 2012 Configuration Manager clients based on the Asset Intelligence hardware inventory reporting classes that you enable.

You can modify the categorization information, which includes product name, vendor, software category, and software family, for inventoried software only at the top-level site in your hierarchy. After you modify the categorization information for predefined software, the validation state for the software changes from Validated to User Defined.

Parameters

-DisableReportingClass<ClassNameType[]>

Specifies an array of Asset Intelligence reporting classes to disable. Valid values are:

- SMS_AutoStartSoftware
- SMS_BrowserHelperObject
- SMS_InstalledExecutable

- SMS_InstalledSoftware
- SMS_SoftwareShortcut
- SMS_SoftwareTag
- SMS_SystemConsoleUsage
- SMS_SystemConsoleUser
- SoftwareLicensingProduct
- SoftwareLicensingService
- Win32_USBDevice

The acceptable values for this parameter are:

SMS_AutoStartSoftware	
SMS_BrowserHelperObject	
SMS_InstalledExecutable	
SMS_InstalledSoftware	
SMS_SoftwareShortcut	
SMS_SoftwareTag	
SMS_SystemConsoleUsage	
SMS_SystemConsoleUser	
SoftwareLicensingProduct	
SoftwareLicensingService	
Win32_USBDevice	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnableAllReportingClass

Indicates that all Asset Intelligence reporting classes are enabled.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnableReportingClass<ClassNameType[]>

Specifies an array of Asset Intelligence reporting classes to enable. Valid values are:

- SMS_AutoStartSoftware
- SMS_BrowserHelperObject
- SMS_InstalledExecutable
- SMS_InstalledSoftware
- SMS_SoftwareShortcut
- SMS_SoftwareTag
- SMS_SystemConsoleUsage
- SMS_SystemConsoleUser
- SoftwareLicensingProduct
- SoftwareLicensingService
- Win32_USBDevice

The acceptable values for this parameter are:

SMS_AutoStartSoftware	
SMS_BrowserHelperObject	
SMS_InstalledExecutable	
SMS_InstalledSoftware	
SMS_SoftwareShortcut	
SMS_SoftwareTag	
SMS_SystemConsoleUsage	
SMS_SystemConsoleUser	
SoftwareLicensingProduct	
SoftwareLicensingService	
Win32_USBDevice	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Change the Asset Intelligence hardware inventory reporting classes

This command enables the reporting class named SMS_InstalledExecutable and disables the reporting class named MS_InstalledSoftware.

```
PS C:\> Set-CMAAssetIntelligenceClass -EnableReportingClassName SMS_InstalledExecutable -  
DisableReportingClassName MS_InstalledSoftware
```

Example 2: Enable all Asset Intelligence hardware inventory reporting classes

This command enables all the Asset Intelligence hardware inventory reporting classes.

```
PS C:\> Set-CMAAssetIntelligenceClass -EnableAllReportingClass
```

Related topics

[Send-CMAAssetIntelligenceCatalogUpdateRequest](#)

[Sync-CMAAssetIntelligenceCatalog](#)

Set-CMAssetIntelligenceSynchronizationPoint

Set-CMAssetIntelligenceSynchronizationPoint

Enables or disables an Asset Intelligence synchronization point.

Syntax

Parameter Set: SetByName

```
Set-CMAssetIntelligenceSynchronizationPoint -SiteSystemServerName <String[]> [-Enabled <Boolean> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetByValueMandatory

```
Set-CMAssetIntelligenceSynchronizationPoint -InputObject <IResultObject> [-Enabled <Boolean> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Set-CMAssetIntelligenceSynchronizationPoint** cmdlet enables or disables one or more Asset Intelligence synchronization points. You must enable the Asset Intelligence synchronization point to perform scheduled Asset Intelligence catalog synchronizations with System Center Online.

Parameters

-Enabled<Boolean>

Indicates whether the Asset Intelligence synchronization point is enabled.

Aliases	ProxyEnabled
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies an Asset Intelligence synchronization point object. To obtain a **CMAssetIntelligenceSynchronizationPoint** object, use the **Get-
CMAssetIntelligenceSynchronizationPoint** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteSystemServerName<String[]>

Specifies an array of fully qualified domain names (FQDN) of the servers that host the site system role.

Aliases	ProxyName
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Enable an Asset Intelligence synchronization point

This command enables the Asset Intelligence synchronization point on the site server named CMDIV-WEST04.CORP.CONTOSO.COM.

```
PS C:\> Set-CMAssetIntelligenceSynchronizationPoint -SiteSystemServerName "CMDIV-  
WEST04.CORP.CONTOSO.COM" -Enabled $True
```

Related topics

[Get-CMAssetIntelligenceSynchronizationPoint](#)

[Add-CMAssetIntelligenceSynchronizationPoint](#)

[Remove-CMAssetIntelligenceSynchronizationPoint](#)

Set-CMAssignedSite

Set-CMAssignedSite

Assigns a client computer to a primary site.

Syntax

Parameter Set: SearchByNameMandatory

```
Set-CMAssignedSite -DeviceName <String[]> -SiteCode <String> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Set-CMAssignedSite -DeviceId <String[]> -SiteCode <String> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Set-CMAssignedSite -InputObject <IResultObject> -SiteCode <String> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Detailed Description

The **Set-CMAssignedSite** cmdlet assigns a client computer to a primary site. When you install a client agent, the installation determines the primary site for the client. This cmdlet assigns a client to a different primary site.

Parameters

-DeviceId<String[]>

Specifies an array of device IDs.

Aliases	ResourceId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeviceName<String[]>

Specifies an array of device names.

Aliases	Name
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a **CMDevice** object. To obtain a **CMDevice** object, use the **Get-CMDevice** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteCode<String>

Specifies the site code of the new primary site to assign the client to.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Reassign a client to a site

This command reassigns the client that has the device ID 2097152000 to the site that has the site code CM7.

```
PS C:\> Set-CMAssignedSite -DeviceId "2097152000" -SiteCode "CM7"
```

Related topics

[Get-CMDevice](#)

Set-CMBaseline

Set-CMBaseline

Changes the settings of configuration baselines.

Syntax

Parameter Set: SetByIdMandatory

```
Set-CMBaseline -Id <String[]> [-AddCategory <String[]> ] [-AddOptionalConfigurationItems <String[]> ] [-AddOSConfigurationItems <String[]> ] [-AddProhibitedConfigurationItems <String[]> ] [-AddRequiredConfigurationItems <String[]> ] [-Description <String> ] [-DesiredConfigurationDigestPath <String> ] [-NewName <String> ] [-RemoveCategory <String[]> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetByNameMandatory

```
Set-CMBaseline -Name <String[]> [-AddCategory <String[]> ] [-AddOptionalConfigurationItems <String[]> ] [-AddOSConfigurationItems <String[]> ] [-AddProhibitedConfigurationItems <String[]> ] [-AddRequiredConfigurationItems <String[]> ] [-Description <String> ] [-DesiredConfigurationDigestPath <String> ] [-NewName <String> ] [-RemoveCategory <String[]> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetByValueMandatory

```
Set-CMBaseline -InputObject <IResultObject> [-AddCategory <String[]> ] [-AddOptionalConfigurationItems <String[]> ] [-AddOSConfigurationItems <String[]> ] [-AddProhibitedConfigurationItems <String[]> ] [-AddRequiredConfigurationItems <String[]> ] [-Description <String> ] [-DesiredConfigurationDigestPath <String> ] [-NewName <String> ] [-RemoveCategory <String[]> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetSecurityScopeById

```
Set-CMBaseline -Id <String[]> -SecurityScopeAction {AddMembership | RemoveMembership} -SecurityScopeName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetSecurityScopeByName

```
Set-CMBaseline -Name <String[]> -SecurityScopeAction {AddMembership | RemoveMembership} -SecurityScopeName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetSecurityScopeByValue

```
Set-CMBaseline -InputObject <IResultObject> -SecurityScopeAction {AddMembership | RemoveMembership} -SecurityScopeName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Set-CMBaseline** cmdlet changes the settings of one or more configuration baselines in Microsoft System Center 2012 Configuration Manager.

Parameters

-AddCategory<String[]>

Specifies an array of names of configuration categories to add to the configuration baselines.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AddOptionalConfigurationItems<String[]>

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AddOSConfigurationItems<String[]>

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AddProhibitedConfigurationItems<String[]>

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AddRequiredConfigurationItems<String[]>

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Description<String>

Specifies a description of the configuration baseline.

Aliases	LocalizedDescription
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-DesiredConfigurationDigestPath<String>

Specifies a path to the configuration data stored as a digest.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Id<String[]>

Specifies an array of IDs of configuration baselines.

Aliases	CId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a **CMBaseline** object. To obtain a **CMBaseline** object, use the **Get-CMBaseline** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-Name<String[]>

Specifies an array of names of configuration baselines.

Aliases	LocalizedDisplayName
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-NewName<String>

Specifies a new name for the configuration baseline.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RemoveCategory<String[]>

Specifies an array of names of configuration categories to remove from the configuration baselines.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-SecurityScopeAction<SecurityScopeActionType>

Specifies an action for a security scope. Valid values are: AddMembership and RemoveMembership.

The acceptable values for this parameter are:

AddMembership	
RemoveMembership	

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SecurityScopeName<String>

Specifies the name of a security scope. A security scope name can be Default or the name of a custom security scope.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Add a membership to a security scope of a configuration baseline

This command adds membership to the security scope named SecScope02 for the configuration baseline named BLineContoso01.

```
PS C:\> Set-CMBaseline -SecurityScopeAction AddMembership -SecurityScopeName "SecScope02" -
Name "BLineContoso01"
```

Example 2: Remove membership from a security scope of a configuration baseline

This command removes membership to the security scope named SecScope02 for the configuration baseline named BLineContoso01.

```
PS C:\> Set-CMBaseline -SecurityScopeAction RemoveMembership -SecurityScopeName "SecScope02"  
-Name "BLineContoso01"
```

Related topics

[Disable-CMBaseline](#)

[Enable-CMBaseline](#)

[Export-CMBaseline](#)

[Get-CMBaseline](#)

[Import-CMBaseline](#)

[New-CMBaseline](#)

[Remove-CMBaseline](#)

[Get-CMBaselineXMLDefinition](#)

[Get-CMBaselineSummarizationSchedule](#)

Set-CMBaselineDeployment

Set-CMBaselineDeployment

Changes settings for a Configuration Manager baseline deployment.

Syntax

Parameter Set: SetBaselineDeploymentByIdMandatory

```
Set-CMBaselineDeployment -BaselineId <String> -CollectionName <String> [-EnableEnforcement <Boolean> ] [-GenerateAlert <Boolean> ] [-MonitoredByScom <Boolean> ] [-OverrideServiceWindow <Boolean> ] [-ParameterValue <Int32> ] [-PostponeDate <DateTime> ] [-PostponeTime <DateTime> ] [-Schedule <IResultObject> ] [-Confirm] [-WhatIf] [<CommonParameters>]
```

Parameter Set: SetBaselineDeploymentByNameMandatory

```
Set-CMBaselineDeployment -BaselineName <String> -CollectionName <String> [-EnableEnforcement <Boolean> ] [-GenerateAlert <Boolean> ] [-MonitoredByScom <Boolean> ] [-OverrideServiceWindow <Boolean> ] [-ParameterValue <Int32> ] [-PostponeDate <DateTime> ] [-PostponeTime <DateTime> ] [-Schedule <IResultObject> ] [-Confirm] [-WhatIf] [<CommonParameters>]
```

Parameter Set: SetBaselineDeploymentByValueMandatory

```
Set-CMBaselineDeployment -Baseline <IResultObject> -CollectionName <String> [-EnableEnforcement <Boolean> ] [-GenerateAlert <Boolean> ] [-MonitoredByScom <Boolean> ] [-OverrideServiceWindow <Boolean> ] [-ParameterValue <Int32> ] [-PostponeDate <DateTime> ] [-PostponeTime <DateTime> ] [-Schedule <IResultObject> ] [-Confirm] [-WhatIf] [<CommonParameters>]
```

Detailed Description

The **Set-CMBaselineDeployment** cmdlet changes settings for a Microsoft System Center 2012 Configuration Manager baseline configuration deployment. A baseline defines the configuration of a product or system established at a specific time. Baselines contain a defined set of required configurations and associated rules. System Center 2012 Configuration Manager assigns baselines to computer in collections, together with a compliance evaluation schedule.

Use the baseline and the name of a collection to specify a deployment to modify. You can specify a baseline by its name or ID, or use the **Get-CMBaseline** cmdlet to get a baseline object.

You can use the **Start-CMBaselineDeployment** cmdlet to begin a deployment.

Parameters

-Baseline<IResultObject>

Specifies a baseline object. To obtain a baseline object, use the **Get-CMBaseline** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-BaselineId<String>

Specifies the ID of a baseline.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-BaselineName<String>

Specifies the name of a baseline.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CollectionName<String>

Specifies the name of a collection. The deployment applies to this collection.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnableEnforcement<Boolean>

Specifies whether to enable enforcement for the baseline. During enforcement, a client reports compliance information about the configurations in a baseline.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-GenerateAlert<Boolean>

Specifies whether Configuration Manager generates alerts during the deployment.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-MonitoredByScom<Boolean>

Specifies whether to apply System Center 2012 – Operations Manager monitoring criteria during the deployment.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OverrideServiceWindow<Boolean>

Specifies whether to override service windows while deploying policies. Service windows are periods of time allocated for maintenance.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ParameterValue<Int32>

Specifies an integer value. This is the parameter value.

Aliases	none
Required?	false
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PostponeDate<DateTime>

Specifies a date, as a **DateTime** object. To obtain a **DateTime** object, use the **Get-Date** cmdlet. For more information, type `Get-Help Get-Date`. This is the date for the deployment if it is postponed.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PostponeTime<DateTime>

Specifies a time, as a **DateTime** object. To obtain a **DateTime** object, use the **Get-Date** cmdlet. This is the time for the deployment if it is postponed.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Schedule<IResultObject>

Specifies a schedule object. This is the schedule for deploying a baseline.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Change whether a deployment generates alerts

This command changes a deployment for the baseline named Baseline 2012 for a collection named All Computers. This command sets the *GenerateAlert* parameter to `$False`.

```
PS C:\> Set-CMBaselineDeployment -BaselineName "Baseline 2012" -CollectionName "All  
Computers" -GenerateAlert $False
```

Example 2: Change deployment settings

This command changes a deployment for the baseline named Baseline A3 for a collection named TSQA Computers. The command specifies values for generation of alerts and Operations Manager monitoring. It also includes as a parameter value and postpone date and time.

```
PS C:\> Set-CMBaselineDeployment -BaselineName "Baseline A3" -CollectionName "TSQA  
Computers" -GenerateAlert $True -MonitoredByScom $True -ParameterValue 60 -PostponeDate  
2013/02/12 -PostponeTime 12:34
```

Related topics

[Start-CMBaselineDeployment](#)

[Get-CMBaseline](#)

Set-CMBaselineSummarizationSchedule

Set-CMBaselineSummarizationSchedule

Configures the summarization schedule for configuration baseline data.

Syntax

Parameter Set: Set

```
Set-CMBaselineSummarizationSchedule -Interval <Int32> [-Unit {Days | Hours | Minutes} ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Set-CMBaselineSummarizationSchedule** cmdlet configures the schedule by which the configuration baseline data in the Microsoft System Center 2012 Configuration Manager is updated with the latest information from the site database.

Parameters

-Interval<Int32>

Specifies an amount of time, as an integer. This value works with the unit type you specify in the *Unit* parameter. Valid values for this parameter depend on the unit that you select:

-- Minutes: 10 through 59.

-- Hours: 1 through 23.

-- Days: 1 through 31.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Unit<SummarizationScheduleUnit>

Specifies a unit to use to define an interval for the summarization schedule. Valid values are:

- Days
- Hours
- Minutes

The acceptable values for this parameter are:

Days	
Hours	
Minutes	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Set the configuration baseline update schedule

This command schedules Configuration Manager to automatically update the configuration baseline data every six hours.

```
PS C:\> Set-CMBaselineSummarizationSchedule -Interval 6 -Unit "Hours"
```

Related topics

[Get-CMBaselineSummarizationSchedule](#)

[Invoke-CMBaselineSummarization](#)

Set-CMBootImage

Set-CMBootImage

Modifies an operating system boot image.

Syntax

Parameter Set: SetById

```
Set-CMBootImage -Id <String[]> [-Description <String> ] [-NewName <String> ] [-PackageSourcePath <String> ] [-SecuredScopeNames <String> ] [-Version <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetByName

```
Set-CMBootImage -Name <String> [-Description <String> ] [-NewName <String> ] [-PackageSourcePath <String> ] [-SecuredScopeNames <String> ] [-Version <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetByValueMandatory

```
Set-CMBootImage -InputObject <IResultObject> [-Description <String> ] [-NewName <String> ] [-PackageSourcePath <String> ] [-SecuredScopeNames <String> ] [-Version <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetSecurityScopeById

```
Set-CMBootImage -Id <String[]> -SecurityScopeAction {AddMembership | RemoveMembership} -SecurityScopeName <String> [-SecuredScopeNames <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetSecurityScopeByName

```
Set-CMBootImage -Name <String> -SecurityScopeAction {AddMembership | RemoveMembership} -SecurityScopeName <String> [-SecuredScopeNames <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetSecurityScopeByValue

```
Set-CMBootImage -InputObject <IResultObject> -SecurityScopeAction {AddMembership | RemoveMembership} -SecurityScopeName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Set-CMBootImage** cmdlet modifies an operating system boot image. Boot images are Windows Preinstallation Environment (Windows PE) images into which you boot a client computer before you install an operating system.

You can add device drivers to a boot image or change its properties. Before you can add a new device driver, you must first import the driver to the Microsoft System Center 2012 Configuration Manager driver catalog and enable it.

A modification to the boot image does not change its source package.

Parameters

-Description<String>

Describes the contents of a boot image.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Id<String[]>

Specifies an array of boot image identifiers.

Aliases	Packageld
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a boot image object.

Aliases	none
Required?	true

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies a name of a boot image.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-NewName<String>

Specifies a new name for the boot image.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PackageSourcePath<String>

Modifies the location of the Windows Imaging (WIM) format file for a boot image.

Aliases	none
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SecuredScopeNames<String>

Specifies a secured scope name.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SecurityScopeAction<SecurityScopeActionType>

Specifies an action for a security scope. Valid values are: AddMembership and RemoveMembership.

The acceptable values for this parameter are:

AddMembership	
RemoveMembership	

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SecurityScopeName<String>

Specifies the name of a security scope. A security scope name can be Default or the name of a custom security scope.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Version<String>

Specifies the version of the boot image.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Rename a boot image object that is identified by using its ID

This command retrieves a boot image by using its ID, and then renames the boot image. Depending on replication issues, this modification can take a long time to display in the Configuration Manager console.

```
PS C:\> Set-CMBootImage -Id "CM100004" -NewName "Windows8 (x64)"
```

Example 2: Rename a boot image object that is identified by using its name

This command retrieves a boot image by using its name, and then renames the boot image. The command also adds a version and description to the boot image object. Depending on replication issues, this modification can take a long time to display in the Configuration Manager console.

```
PS C:\> Set-CMBootImage -Name "Boot Image (x64)" -NewName "Windows 8 x64" -Version "6.2.8400.1" -Description "Microsoft Windows 8 PE (x64)"
```

Related topics

[Get-CMBootImage](#)

[New-CMBootImage](#)

[Remove-CMBootImage](#)



Set-CMBoundary

Set-CMBoundary

Modifies boundary settings.

Syntax

Parameter Set: SetById

```
Set-CMBoundary -Id <String[]> [-NewName <String> ] [-Type {ADSite | IPRange | IPSubnet | IPv6Prefix} ] [-Value <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetByName

```
Set-CMBoundary -Name <String[]> [-NewName <String> ] [-Type {ADSite | IPRange | IPSubnet | IPv6Prefix} ] [-Value <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetValueMandatory

```
Set-CMBoundary -InputObject <IResultObject> [-NewName <String> ] [-Type {ADSite | IPRange | IPSubnet | IPv6Prefix} ] [-Value <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Set-CMBoundary** cmdlet modifies boundary settings.

In Microsoft System Center 2012 Configuration Manager, a boundary is an intranet location that contains one or more devices that you can manage. A boundary can be an IP subnet, Active Directory site name, IPv6 prefix, or an IP address range.

Parameters

-Id<String[]>

Specifies an array of boundary identifiers (IDs).

Aliases	BoundaryId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-InputObject<IResultObject>

Specifies an input object to this cmdlet. You can get the input object by using the **Get-CMBoundary** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String[]>

Specifies an array of boundary names.

Aliases	DisplayName
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-NewName<String>

Specifies a new name for a boundary.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Type<BoundaryTypes>

Specifies a boundary type. Valid values are: ADSite, IPV6Prefix, IPSubnet, and IPRange.

The acceptable values for this parameter are:

ADSite	Specifies an Active Directory site.
IPRange	Specifies an IP address range.
IPSubnet	Specifies an IP subnet.
IPV6Prefix	Specifies an IP v6 prefix.

Aliases	BoundaryType
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Value<String>

Specifies the data that describes the boundary. For example, an Active Directory site value can be Default-ADSite.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Rename a boundary

This command changes a boundary name from Default-ADSite to ADSiteBoundary01.

```
PS C:\> Set-CMBoundary -Name "Default-ADSite" -NewName "ADSiteBoundary01"
```

Example 2: Modify the value of a boundary by using an InputObject

In this example, the first command gets a boundary that has the ID of 16777217 and inserts it into the input object \$BoundaryObj.

The second command identifies the boundary by using the input object `$BoundaryObj` and modifies its value to `IPSubnet17`.

```
PS C:\> $BoundaryObj = Get-CMBoundary -Id "16777217"
```

```
PS C:\> Set-CMBoundary -InputObject $BoundaryObj -Value "IPSubnet17"
```

Related topics

[Get-CMBoundary](#)

[New-CMBoundary](#)

[Remove-CMBoundary](#)

Set-CMBoundaryGroup

Set-CMBoundaryGroup

Modifies the properties of a boundary group.

Syntax

Parameter Set: SetById

```
Set-CMBoundaryGroup -Id <String[]> [-DefaultSiteCode <String> ] [-Description <String> ] [-NewName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetByName

```
Set-CMBoundaryGroup -Name <String> [-DefaultSiteCode <String> ] [-Description <String> ] [-NewName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetValueMandatory

```
Set-CMBoundaryGroup -InputObject <IResultObject> [-DefaultSiteCode <String> ] [-Description <String> ] [-NewName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetSecurityScopeById

```
Set-CMBoundaryGroup -Id <String[]> -SecurityScopeAction {AddMembership | RemoveMembership} -SecurityScopeName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetSecurityScopeByName

```
Set-CMBoundaryGroup -Name <String> -SecurityScopeAction {AddMembership | RemoveMembership} -SecurityScopeName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetSecurityScopeByValue

```
Set-CMBoundaryGroup -InputObject <IResultObject> -SecurityScopeAction {AddMembership | RemoveMembership} -SecurityScopeName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Set-CMBoundaryGroup** cmdlet modifies the properties of a boundary group. A boundary group is a collection of boundaries. For more information about boundaries, see [Planning for Boundaries and Boundary Groups in Configuration Manager](http://go.microsoft.com/fwlink/?LinkId=266225) (<http://go.microsoft.com/fwlink/?LinkId=266225>) on TechNet and the **New-CMBoundary** cmdlet.

Parameters

-DefaultSiteCode<String>

Specifies the default site code of a boundary group.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Description<String>

Specifies a description for a boundary group.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Id<String[]>

Specifies an array of identifiers for one or more boundary groups.

Aliases	GroupId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies an input object to this cmdlet. You can get the input object by using the **Get-CMBoundaryGroup** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies a name for a boundary group.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-NewName<String>

Specifies a new name for a boundary group.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-SecurityScopeAction<SecurityScopeActionType>

Specifies an action for a security scope. Valid values are: AddMembership and RemoveMembership.

The acceptable values for this parameter are:

AddMembership	
RemoveMembership	

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SecurityScopeName<String>

Specifies the name of a security scope. A security scope name can be Default or the name of a custom security scope.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Rename a boundary group

This command renames a boundary group.

```
PS C:\> Set-CMBoundaryGroup -Name "BGroup01" -NewName "BGroup00"
```

Example 2: Add a security scope to a boundary group

This command adds the security scope OSDeploymentScope to the boundary group BGroup02.

```
PS C:\> Set-CMBoundaryGroup -SecurityScopeAction AddMembership -SecurityScopeName "OSDeploymentScope" -Name "BGroup02"
```

Related topics

[Get-CMBoundaryGroup](#)

[New-CMBoundaryGroup](#)

[Remove-CMBoundaryGroup](#)

[Set-CMSecurityScope](#)

[New-CMBoundary](#)

Set- CMClientAuthCertificateProfileConfigurationItem

Set-CMClientAuthCertificateProfileConfigurationItem

Modifies a certificate profile.

Syntax

Parameter Set: SetByName

```
Set-CMClientAuthCertificateProfileConfigurationItem -Name <String[]> [-  
DesiredConfigurationDigestPath <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetById

```
Set-CMClientAuthCertificateProfileConfigurationItem -Id <String[]> [-  
DesiredConfigurationDigestPath <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetValue

```
Set-CMClientAuthCertificateProfileConfigurationItem -InputObject <IResultObject> [-  
DesiredConfigurationDigestPath <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Set-CMClientAuthCertificateProfileConfigurationItem** cmdlet modifies a certificate profile. Client computers use certificate profiles to authenticate when they use services such as a virtual private network (VPN) or a wireless network.

Parameters

-DesiredConfigurationDigestPath<String>

Specifies a path to the configuration data stored as a digest.

Aliases	none
Required?	false
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Id<String[]>

Specifies an array of IDs of certificate profiles.

Aliases	CId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a certificate profile object. To obtain a certificate profile object, use the **Get-
CMClientAuthCertificateProfileConfigurationItem** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String[]>

Specifies an array of names of certificate profiles.

Aliases	LocalizedDisplayName
Required?	true

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

[Copy-CMClientAuthCertificateProfileConfigurationItem](#)

[Get-CMClientAuthCertificateProfileConfigurationItem](#)

[New-CMClientAuthCertificateProfileConfigurationItem](#)

[Remove-CMClientAuthCertificateProfileConfigurationItem](#)

Set-CMClientPushInstallation

Set-CMClientPushInstallation

Changes settings of a client push installation.

Syntax

Parameter Set: SearchByNameMandatory

```
Set-CMClientPushInstallation -Name <String> [-ChosenAccount <String[]> ] [-EnableAutomaticClientPushInstallation <Boolean> ] [-EnableSystemTypeConfigurationManager <Boolean> ] [-EnableSystemTypeServer <Boolean> ] [-EnableSystemTypeWorkstation <Boolean> ] [-InstallationProperty <String> ] [-InstallClientToDomainController <Boolean> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchBySiteCodeMandatory

```
Set-CMClientPushInstallation -SiteCode <String> [-ChosenAccount <String[]> ] [-EnableAutomaticClientPushInstallation <Boolean> ] [-EnableSystemTypeConfigurationManager <Boolean> ] [-EnableSystemTypeServer <Boolean> ] [-EnableSystemTypeWorkstation <Boolean> ] [-InstallationProperty <String> ] [-InstallClientToDomainController <Boolean> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Set-CMClientPushInstallation -InputObject <IResultObject> [-ChosenAccount <String[]> ] [-EnableAutomaticClientPushInstallation <Boolean> ] [-EnableSystemTypeConfigurationManager <Boolean> ] [-EnableSystemTypeServer <Boolean> ] [-EnableSystemTypeWorkstation <Boolean> ] [-InstallationProperty <String> ] [-InstallClientToDomainController <Boolean> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Set-CMClientPushInstallation** cmdlet changes the settings of an object that installs a Microsoft System Center 2012 Configuration Manager client by using client push. A client push installation installs client software on computers that System Center 2012 Configuration Manager discovered. When you configure client push installation for a site, the client installation automatically runs on the computers that System Center 2012 Configuration Manager discovered within the site's configured boundaries when those boundaries are part of a boundary group. You can also start a client push installation by running the Client Push Installation Wizard for a specific collection or resource within a collection.

For more information about how to install clients, see [How to Install Clients on Windows-Based Computers in Configuration Manager](http://go.microsoft.com/fwlink/?LinkId=247203) (http://go.microsoft.com/fwlink/?LinkId=247203) on TechNet.

Parameters

-ChosenAccount<String[]>

Specifies an array of accounts for Configuration Manager to use when it connects to the computer to install the client software.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnableAutomaticClientPushInstallation<Boolean>

Indicates whether Configuration Manager automatically uses client push for discovered computers.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnableSystemTypeConfigurationManager<Boolean>

Indicates whether Configuration Manager pushes the client software to Configuration Manager site system servers.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnableSystemTypeServer<Boolean>

Indicates whether Configuration Manager pushes the client software to servers.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnableSystemTypeWorkstation<Boolean>

Indicates whether Configuration Manager pushes the client software to workstations.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a client push installation object. To obtain a client push installation object, use the **Get-CMClientPushInstallation** cmdlet.

Aliases	none
Required?	true
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InstallationProperty<String>

Specifies any installation properties to use when installing the Configuration Manager client.

For System Center 2012 Configuration Manager with no service pack: You can specify only installation properties for the Windows Installer package (Client.msi); you cannot specify properties for CCMSSetup.exe.

For System Center 2012 Configuration Manager SP1: You can specify installation properties for the Windows Installer package (Client.msi) and the following CCMSSetup.exe properties:

- forcereboot
- skipprereq
- logon
- BITSPriority
- downloadtimeout
- forceinstall

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InstallClientToDomainController<Boolean>

Indicates whether to use automatic site-wide client push installation to install the Configuration Manager client software on domain controllers.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies a name for the client push installation.

Aliases	SiteName
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteCode<String>

Specifies the site code of the Configuration Manager site that hosts the site system role.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Change the settings of a client push installation

This command specifies that Configuration Manager automatically uses client push for discovered computers. The command specifies that Configuration Manager pushes the client software to site system servers and uses the account named CENTRAL\00ID\$ to connect to the computer to install the client software. The *InstallationProperty* parameter sets the value of the SMSSITECODE property for the Windows Installer package to CM1. This setting assigns the client to the site that has the site code CM1.

```
PS C:\> Set-CMClientPushInstallation -EnableAutomaticClientPushInstallation $True -  
EnableSystemTypeConfigurationManager $True -ChosenAccount "CENTRAL\00ID$" -  
InstallationProperty "SMSSITECODE=CM1"
```

Related topics

[Get-CMClientPushInstallation](#)

Set-CMClientSetting

Set-CMClientSetting

Changes client settings for Configuration Manager devices and users.

Syntax

Parameter Set: SetBackgroundIntelligentTransferSettingsByName

```
Set-CMClientSetting -Name <String> [-EnableBITSMaxBandwidth <Boolean> ] [-  
EnableDownloadOffSchedule <Boolean> ] [-MaxBandwidthValidFrom <Int32> ] [-  
MaxBandwidthValidTo <Int32> ] [-MaxTransferRateOffSchedule <Int32> ] [-  
MaxTransferRateOnSchedule <Int32> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetByName

```
Set-CMClientSetting -Name <String> [-Description <String> ] [-NewName <String> ] [-Priority  
{Decrease | Increase} ] [-SecuredScopeNames <String> ] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SetClientPolicySettingsByName

```
Set-CMClientSetting -Name <String> [-EnableUserPolicyOnInternet <Boolean> ] [-  
EnableUserPolicyPolling <Boolean> ] [-PolicyPollingInterval <Int32> ] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SetCloudSettingsByName

```
Set-CMClientSetting -Name <String> [-AllowCloudDistributionPoint <Boolean> ] [-Confirm] [-  
WhatIf] [ <CommonParameters>]
```

Parameter Set: SetComplianceSettingsByName

```
Set-CMClientSetting -Name <String> [-EnableComplianceEvaluation <Boolean> ] [-  
EnableUserDataAndProfile <Boolean> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetComputerAgentSettingsByName

```
Set-CMClientSetting -Name <String> [-AddPortalToTrustedSiteList <Boolean> ] [-  
AllowPortalToHaveElevatedTrust <Boolean> ] [-BrandingTitle <String> ] [-  
DisplayNewProgramNotification <Boolean> ] [-EnableThirdPartyOrchestration {No | Yes} ] [-  
FinalReminderMinutesInterval <Int32> ] [-InitialReminderHoursInterval <Int32> ] [-  
InstallRestriction {AllUsers | NoUsers | OnlyAdministrators |  
OnlyAdministratorsAndPrimaryUsers} ] [-InterimReminderHoursInterval <Int32> ] [-PortalUrl  
<String> ] [-PowerShellExecutionPolicy {AllSigned | Bypass | Restricted} ] [-  
SuspendBitLocker {Always | Never} ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetComputerRestartSettingsByName

```
Set-CMClientSetting -Name <String> [-RebootLogoffNotificationCountdownDurationMinutes  
<Int32> ] [-RebootLogoffNotificationFinalWindowMinutes <Int32> ] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SetEndpointProtectionSettingsByName

Set-CMClientSetting -Name <String> [-DisableFirstSignatureUpdate <Boolean>] [-
EnableEndpointProtection <Boolean>] [-ForceRebootPeriod <Int32>] [-
InstallEndpointProtectionClient <Boolean>] [-RemoveThirdParty <Boolean>] [-SuppressReboot
<Boolean>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SetHardwareInventorySettingsByName

Set-CMClientSetting -Name <String> [-EnableHardwareInventory <Boolean>] [-InventoryReportId
<String>] [-InventorySchedule <IResultObject>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SetMeteredNetworksSettingsByName

Set-CMClientSetting -Name <String> [-MeteredNetworkUsage {Allow | Block | Limit | None}] [-
Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SetMobileDeviceSettingsByName

Set-CMClientSetting -Name <String> [-DeviceEnrollmentProfileName <String>] [-
EnableDeviceEnrollment <Boolean>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SetNetworkAccessProtectionSettingsByName

Set-CMClientSetting -Name <String> [-EnableNetworkAccessProtection <Boolean>] [-ForceScan
<Boolean>] [-NapEvaluationSchedule <IResultObject>] [-UseUtcForEvaluationTime <Boolean>]
[-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SetPowerManagementSettingsByName

Set-CMClientSetting -Name <String> [-AllowUserToOptOutFromPowerPlan <Boolean>] [-
EnablePowerManagement <Boolean>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SetRemoteToolsSettingsByName

Set-CMClientSetting -Name <String> [-AccessLevel {FullControl | NoAccess | ViewOnly}] [-
AllowClientChange <Boolean>] [-AllowPermittedViewersToRemoteDesktop <Boolean>] [-
AllowRemoteControlOfUnattendedComputer <Boolean>] [-AudibleSignal {PlayNoSound |
PlaySoundAtBeginAndEnd | PlaySoundRepeatedly}] [-FirewallExceptionProfile {Disabled |
Domain | Private | Public}] [-GrantRemoteControlPermissionToLocalAdministrator <Boolean>]
[-ManageRemoteDesktopSetting <Boolean>] [-ManageSolicitedRemoteAssistance <Boolean>] [-
ManageUnsolicitedRemoteAssistance <Boolean>] [-PermittedViewer <String[]>] [-
PromptUserForPermission <Boolean>] [-RemoteAssistanceAccessLevel {FullControl | None |
RemoteViewing}] [-RequireAuthentication <Boolean>] [-ShowNotificationIconOnTaskbar
<Boolean>] [-ShowSessionConnectionBar <Boolean>] [-Confirm] [-WhatIf] [
<CommonParameters>]

Parameter Set: SetSecurityScopeByName

Set-CMClientSetting -Name <String> -SecurityScopeAction {AddMembership | RemoveMembership} -
SecurityScopeName <String> [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SetSoftwareDeploymentSettingsByName

Set-CMClientSetting -Name <String> [-EvaluationSchedule <IResultObject>] [-Confirm] [-
WhatIf] [<CommonParameters>]

Parameter Set: SetSoftwareInventorySettingsByName

Set-CMClientSetting -Name <String> [-EnableSoftwareInventory <Boolean>] [-
SoftwareInventoryDisplayName <String>] [-SoftwareInventoryFileInventoriedName <String>
] [-SoftwareInventoryFileName <String>] [-SoftwareInventorySchedule <IResultObject>] [-
Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SetSoftwareMeteringSettingsByName
Set-CMClientSetting -Name <String> [-DataCollectionSchedule <IResultObject>] [-
EnableSoftwareMetering <Boolean>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SetSoftwareUpdatesSettingsByName
Set-CMClientSetting -Name <String> [-BatchingTimeout <Int32>] [-
DeploymentEvaluationSchedule <IResultObject>] [-EnableSoftwareUpdatesOnClient <Boolean>]
[-EnforceMandatory <Boolean>] [-ScanSchedule <IResultObject>] [-TimeUnit {Days | Hours}]
[-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SetStateMessagingSettingsByName
Set-CMClientSetting -Name <String> [-StateMessagingReportingCycleMinutes <Int32>] [-
Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SetUserDeviceAffinitySettingsByName
Set-CMClientSetting -Name <String> [-AllowUserAffinity <Boolean>] [-AutoApproveAffinity
<Boolean>] [-UserAffinityLogOnThresholdMinutes <Int32>] [-UserAffinityUsageThresholdDays
<Int32>] [-Confirm] [-WhatIf] [<CommonParameters>]

Detailed Description

The **Set-CMClientSetting** cmdlet changes client settings for Microsoft System Center 2012 Configuration Manager devices and users. System Center 2012 Configuration Manager provides default values for all client settings, but you can use this cmdlet to modify settings objects. Settings objects determine settings for individual clients. For more information about client settings, see [About Client Settings in Configuration Manager](http://go.microsoft.com/fwlink/?LinkId=266226) (http://go.microsoft.com/fwlink/?LinkId=266226) on TechNet.

Client settings for devices include the following categories:

- Client Policy
- Computer Agent
- Computer Restart
- System Center 2012 Endpoint Protection
- Hardware Inventory
- Metered Internet Connections
- Network Access Protection (NAP)
- Power Options
- Remote Tools
- Software Deployment
- Software Inventory
- Software Updates
- User and Device Affinity

Client settings for users include the following categories:

- Mobile Devices

-- User and Device Affinity

To modify a client setting, specify it by name.

Parameters

-AccessLevel<AccessLevelType>

Specifies a level of allowed remote control access. Valid values are:

-- FullControl

-- NoAccess

-- None

-- ViewOnly

The acceptable values for this parameter are:

FullControl	
NoAccess	
ViewOnly	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AddPortalToTrustedSiteList<Boolean>

Specifies whether to add the default Application Catalog website to the Internet Explorer trusted sites zone.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AllowClientChange<Boolean>

Specifies whether users can change policy or notification settings in Software Center.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AllowCloudDistributionPoint<Boolean>

Specifies whether a device or user can access content from a cloud-based distribution point.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AllowPermittedViewersToRemoteDesktop<Boolean>

Specifies whether to allow members of the remote tools Permitted viewers list to initiate remote desktop connections from the Configuration Manager console.

Aliases	none
Required?	false
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AllowPortalToHaveElevatedTrust<Boolean>

Specifies whether to allow a portal to have elevated trust.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AllowRemoteControlOfUnattendedComputer<Boolean>

Specifies whether to allow remote control of a computer with no user logged onto that computer.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AllowUserAffinity<Boolean>

Specifies whether the client automatically configures user device affinity from usage data.

Aliases	none
Required?	false
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AllowUserToOptOutFromPowerPlan<Boolean>

Specifies whether to allow users to exclude a device from power management settings.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AudibleSignal<AudibleSignalType>

Specifies what kind of sound a client computer plays while under remote control. This setting does not apply to remote assistance. Valid values are:

- None
- PlayNoSound
- PlaySoundAtBeginAndEnd
- PlaySoundRepeatedly

The acceptable values for this parameter are:

PlayNoSound	
PlaySoundAtBeginAndEnd	
PlaySoundRepeatedly	

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AutoApproveAffinity<Boolean>

Specifies whether users can define their primary devices.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-BatchingTimeout<Int32>

Specifies a timeout value, as an integer. Specify a value of Hours or Days by using the *TimeUnit* parameter. When an update deadline passes, Configuration Manager deploys all updates pending within this period.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-BrandingTitle<String>

Specifies a Configuration Manager branding title. This branding information helps users identify Configuration Manager as a trusted source.

Aliases	none
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DataCollectionSchedule<IResultObject>

Specifies a data collection schedule object. To obtain a schedule object, use the **New-CMSchedule** cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeploymentEvaluationSchedule<IResultObject>

Specifies a deployment evaluation schedule as a schedule object. To obtain a schedule object, use the **New-CMSchedule** cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Description<String>

Specifies a description for client settings.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeviceEnrollmentProfileName<String>

Specifies a name for an enrollment profile for mobile devices. That profile contains information about the certificate template to use during the enrollment process, the site that contains an enrollment point and enrollment proxy point, and the site that manages a device after enrollment.

In order to specify a profile, provide a value of \$True for the *EnableDeviceEnrollment* parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DisableFirstSignatureUpdate<Boolean>

Specifies whether to disable the first signature update on client from a remote source.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DisplayNewProgramNotification<Boolean>

Specifies whether Configuration Manager shows the user notifications for software availability or software installations. If this parameter has a value of \$False, the user sees only restart notifications.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnableBITSMaxBandwidth<Boolean>

Specifies whether to enable maximum bandwidth for Background Intelligent Transfer Service (BITS) background transfers.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnableComplianceEvaluation<Boolean>

Specifies whether to enable compliance evaluation for this client.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnableDeviceEnrollment<Boolean>

Specifies whether to allow users to enroll mobile devices.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnableDownloadOffSchedule<Boolean>

Specifies whether allow BITS downloads outside of a throttling window.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnableEndpointProtection<Boolean>

Specifies whether to manage existing Endpoint Protection clients. Specify a value of \$True if you already installed an Endpoint Protection client and want to manage it with Configuration Manager. Specify a value of \$True if you want to create a script to uninstall an existing antimalware solution, install the Endpoint Protection client, and deploy this script by using a Configuration Manager application or package.

Aliases	none
Required?	false
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnableHardwareInventory<Boolean>

Specifies whether to enable hardware inventory for a client.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnableNetworkAccessProtection<Boolean>

Specifies whether to enable NAP for a client. NAP requires a System Health Validator point for the Configuration Manager site in order to enforce policies or restrict access.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnablePowerManagement<Boolean>

Specifies whether to enable power management for a client. When enabled, the power management client agent sends information about power capabilities and usage to Configuration Manager. You can apply power management plans from Configuration Manager to clients.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnableSoftwareInventory<Boolean>

Indicates whether to enable software inventory for a client.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnableSoftwareMetering<Boolean>

Specifies whether to enable software metering for a client.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnableSoftwareUpdatesOnClient<Boolean>

Indicates whether to enable software updates for a client.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-

EnableThirdPartyOrchestration<EnableThirdPartyOrchestrationType>

Specifies whether Software Updates and Software Distribution agents wait for third-party software to install updates and applications.

The acceptable values for this parameter are:

No	
Yes	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnableUserDataAndProfile<Boolean>

Specifies whether to enable user data and profile settings.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnableUserPolicyOnInternet<Boolean>

Specifies whether users receive a user policy when logged on to a computer on the Internet. In order for users to receive user policy, you must enable user polling. You can use the *EnableUserPolicyPolling* parameter to enable user polling. An Internet-based management point must authenticate the user.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnableUserPolicyPolling<Boolean>

Specifies whether, when Configuration Manager discovers a user, Configuration Manager clients on computers receive applications and programs targeted for that user.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnforceMandatory<Boolean>

Indicates whether to enforce all mandatory software update deployments that have deadlines within a specified period of time.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EvaluationSchedule<IResultObject>

Specifies an evaluation schedule object. To obtain a schedule object, use the **New-CMSchedule** cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-FinalReminderMinutesInterval<Int32>

Specifies an interval, in minutes, between the final warning of a required restart and the restart. Some software updates require a computer restart. The computer warns the user before the computer restarts.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-FirewallExceptionProfile<FirewallExceptionProfileType[]>

Specifies a firewall exception profile. Valid values are:

- Disabled
- Domain
- Private
- Public

The acceptable values for this parameter are:

Disabled	
Domain	
Private	
Public	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ForceRebootPeriod<Int32>

Specifies a period, in hours, between forced computer restarts.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ForceScan<Boolean>

Specifies whether to enable force scan.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-GrantRemoteControlPermissionToLocalAdministrator<Boolean>

Specifies whether local administrators on the server initiating a remote control connection can establish remote control sessions to this client.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InitialReminderHoursInterval<Int32>

Specifies an interval, in hours, for the initial reminder.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InstallEndpointProtectionClient<Boolean>

Specifies whether to install and enable the Endpoint Protection client on this client if it is not already installed.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InstallRestriction<InstallRestrictionType>

Specifies which users can initiate an install. Valid values are:

- AllUsers
- NoUsers
- OnlyAdministrators
- OnlyAdministratorsAndPrimaryUsers

The acceptable values for this parameter are:

AllUsers	
NoUsers	
OnlyAdministrators	
OnlyAdministratorsAndPrimaryUsers	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InterimReminderHoursInterval<Int32>

Specifies an interval, in hours, for an interim reminder.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InventoryReportId<String>

Specifies an inventory report ID.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InventorySchedule<IResultObject>

Specifies an inventory schedule object. To obtain a schedule object, use the **New-CMSchedule** cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ManageRemoteDesktopSetting<Boolean>

Specifies whether to allow Configuration Manager to manage Remote Desktop sessions for computers.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ManageSolicitedRemoteAssistance<Boolean>

Specifies whether to allow Configuration Manager to manage solicited remote assistance sessions.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ManageUnsolicitedRemoteAssistance<Boolean>

Specifies whether to allow Configuration Manager to manage unsolicited remote assistance sessions.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-MaxBandwidthValidFrom<Int32>

Specifies an integer value for maximum bandwidth valid from.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-MaxBandwidthValidTo<Int32>

Specifies an integer value for maximum bandwidth valid to.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-MaxTransferRateOffSchedule<Int32>

Specifies an integer value for maximum transfer rate off schedule.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-MaxTransferRateOnSchedule<Int32>

Specifies an integer value for maximum transfer rate on schedule.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-MeteredNetworkUsage<MeteredNetworkUsageType>

Specifies a type of metered network usage to allow. Valid values are:

- Allow
- Block
- Limit
- None

The acceptable values for this parameter are:

Allow	
Block	
Limit	
None	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies a name for a client setting.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-NapEvaluationSchedule<IResultObject>

Specifies a NAP evaluation schedule as a schedule object. To obtain a schedule object, use the **New-CMSchedule** cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-NewName<String>

Specifies a new name for a client setting.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PermittedViewer<String[]>

Specifies an array of names of users who can establish remote control sessions to a client computer.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PolicyPollingInterval<Int32>

Specifies how frequently, in minutes, client computers download client policy.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PortalUrl<String>

Specifies a URL for a portal for a client.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PowerShellExecutionPolicy<PowerShellExecutionPolicyType>

Specifies how Configuration Manager runs Windows PowerShell scripts on remote computers. Valid values are

- AllSigned
- Bypass
- Restricted

The default value is Restricted.

When you select Restricted, the Configuration Manager client uses the current Windows PowerShell configuration on the client computer, which determines whether unsigned scripts run.

The acceptable values for this parameter are:

AllSigned	
Bypass	
Restricted	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Priority<PriorityChangeType>

Specifies a priority change for a client setting. Valid values are: Decrease and Increase.

The acceptable values for this parameter are:

Decrease	
Increase	

Aliases	none
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PromptUserForPermission<Boolean>

Specifies whether a client computer displays a message asking for user permission before it allows a remote control session.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RebootLogoffNotificationCountdownDurationMinutes<Int32>

Specifies a length of time, in minutes, for a logoff notification countdown.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RebootLogoffNotificationFinalWindowMinutes<Int32>

Specifies a length of time, in minutes, for a logoff final notification window.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-

RemoteAssistanceAccessLevel<RemoteAssistanceAccessLevelType>

Specifies a level of access to assign to remote assistance sessions initiated in Configuration Manager. A user at the client computer always grants permission for a remote assistance session to occur. Valid values are:

- FullControl
- None
- RemoteViewing

The acceptable values for this parameter are:

FullControl	
None	
RemoteViewing	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RemoveThirdParty<Boolean>

Specifies whether to remove third party.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RequireAuthentication<Boolean>

Specifies whether to use network-level authentication to establish Remote Desktop connections to a client computer.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ScanSchedule<IResultObject>

Specifies a scan schedule as a schedule object. To obtain a schedule object, use the **New-CMSchedule** cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SecuredScopeNames<String>

Specifies a name of security scope for a client setting.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SecurityScopeAction<SecurityScopeActionType>

Specifies an action for a security scope. Valid values are: AddMembership and RemoveMembership.

The acceptable values for this parameter are:

AddMembership	
RemoveMembership	

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SecurityScopeName<String>

Specifies the name of a security scope. A security scope name can be Default or the name of a custom security scope.

Aliases	none
Required?	true
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ShowNotificationIconOnTaskbar<Boolean>

Specifies whether to display an icon on the taskbar of a client computer to indicate an active remote control session.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ShowSessionConnectionBar<Boolean>

Specifies whether to display a high-visibility session connection bar on a client computer to indicate an active remote control session.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SoftwareInventoryFileDisplayName<String>

Specifies the display name to use in place of an inventoried name specified by the *SoftwareInventoryFileInventoriedName* parameter. This parameter allows you to standardize inventory information for software names that differ in different file headers.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SoftwareInventoryFileInventoriedName<String>

Specifies an inventoried manufacturer or product name. During software inventory, Configuration Manager retrieves inventoried names from header information on client devices.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SoftwareInventoryFileName<String>

Specifies a name for the file you want to collect during inventory. You can use the wildcard (*) to represent any string of text and the question mark (?) to represent any single character.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SoftwareInventorySchedule<IResultObject>

Specifies an inventory schedule as a schedule object. To obtain a schedule object, use the **New-CMSchedule** cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-StateMessagingReportingCycleMinutes<Int32>

Specifies a length of time, in minutes, for a reporting cycle for state messaging.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SuppressReboot<Boolean>

Specifies whether to bypass a required computer restart after installing the System Center 2012 Endpoint Protection client.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SuspendBitLocker<SuspendBitLockerType>

Specifies whether to bypass a required BitLocker Drive Encryption PIN entry when a computer restarts after a software installation. This setting applies only when Configuration Manager initiates a restart.

Valid values are:

- Always. Configuration Manager temporarily suspends the BitLocker requirement to enter a PIN.
- Never. Configuration Manager does not suspend the BitLocker requirement to enter a PIN on the next computer startup after it has installed software that requires a restart.

If you select Never, the software installation cannot finish until the user enters the PIN to complete the standard startup process.

The acceptable values for this parameter are:

Always	
Never	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-TimeUnit<BatchingTimeoutType>

Specifies the unit for the value specified in the *BatchingTimeout* parameter. Valid values are: Hours and Days.

The acceptable values for this parameter are:

Days	
Hours	

Aliases	none
Required?	false
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UserAffinityLogOnThresholdMinutes<Int32>

Specifies the number of minutes before Configuration Manager creates a user device affinity mapping. For example, if you set the *UserAffinityLogOnThresholdMinutes* parameter to 60 minutes and the *UserAffinityUsageThresholdDays* parameter to 5 days, if a user uses a device for 60 minutes over a period of 5 days, Configuration Manager creates a user device affinity.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UserAffinityUsageThresholdDays<Int32>

Specifies the number of days over which to measure the usage based affinity threshold. For example, if you set the *UserAffinityLogOnThresholdMinutes* parameter to 60 minutes and the *UserAffinityUsageThresholdDays* parameter to 5 days, if a user uses a device for 60 minutes over a period of 5 days, Configuration Manager creates a user device affinity.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UseUtcForEvaluationTime<Boolean>

Specifies whether to use Coordinated Universal Time (UTC), also known as Greenwich Mean Time, to configure a recurring interval. If you specify `$False`, Configuration Manager uses local time.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Rename a client setting

This command renames the client setting object. The new name is Client Settings TSQA. The command also adds a description for the client setting object.

```
PS C:\> Set-CMClientSetting -Name "Client Settings Main" -Description "Client settings for TSQA office site." -NewName "Client Settings TSQA"
```

Example 2: Configure power management

This command allows users to opt out of power plans and disables power management for the clients with the setting named TSQA02.

```
PS C:\> Set-CMClientSetting -Name "TSQA02" -AllowUserToOptOutFromPowerPlan $True -EnablePowerManagement $False
```

Example 3: Set state messaging reporting cycle value

This command sets a state messaging reporting cycle value of 10 minutes.

```
PS C:\> Set-CMClientSetting -Name "TSQA02" -StateMessagingReportingCycleMinutes 10
```

Example 4: Configure user affinity

This command configures user affinity settings for a client setting named TSQA03. The command disables auto approval of affinity. The command sets the *UserAffinityLogOnThresholdMinutes* parameter to 2800 minutes and the *UserAffinityUsageThresholdDays* parameter to 20 days, so if a user uses a device for 2800 minutes over a period of 20 days, Configuration Manager creates a user device affinity.

```
PS C:\> Set-CMClientSetting -Name "TSQA03" -AutoApproveAffinity $False -UserAffinityLogOnThresholdMinutes 2800 -UserAffinityUsageThresholdDays 20
```

Example 5: Allow user affinity

This command changes the client setting named TSQA04 to have a client automatically configure user device affinity from usage data.

```
PS C:\> Set-CMClientSetting -Name "TSQA04" -AllowUserAffinity $True
```

Example 6: Set bandwidth for client

This command changes settings for the client settings object named TSQA05. The command enables maximum bandwidth for BITS transfers and enables off schedule downloads. The command also specifies values for maximum bandwidth value from and to and maximum transfer rate on schedule.

```
PS C:\> Set-CMClientSetting -Name "TSQA05" -EnableBITSMaxBandwidth $True  
EnableDownloadOffSchedule $True -MaxBandwidthValidFrom 8 -MaxBandwidthValidTo 15 -  
MaxTransferRateOnSchedule 1500
```

Example 7: Configure user policies on the Internet

This command changes settings for the client settings object named TSQA06. The command enables user policy on the Internet, enables user policy polling, and sets a policy polling interval.

```
PS C:\> Set-CMClientSetting -Name "TSQA06" -EnableUserPolicyOnInternet $True -  
EnableUserPolicyPolling $False -EnableUserPolicyOnInternet $True -PolicyPollingInterval 50
```

Example 8: Disable compliance evaluation

This command disables compliance evaluation for the setting named TSQA07.

```
PS C:\> Set-CMClientSetting -Name "TSQA07" -EnableComplianceEvaluation $False
```

Example 9: Set computer agent settings

This command changes settings for the client settings object named TSQA09. The command specifies a portal and adds that portal to the trusted site list and allows it to have elevated trust. The command specifies a branding title, Contoso IT. The command enables third party orchestration. The command sets final reminder and initial reminder intervals. The command also specifies that only administrators can install software, selects Bypass as the Windows PowerShell execution policy, and suspends a BitLocker PIN requirement.

```
PS C:\> Set-CMClientSetting -Name "TSQA09" -AddPortalToTrustedSiteList $True -  
AllowPortalToHaveElevatedTrust $True -BrandingTitle "Contoso IT" -  
EnableThirdPartyOrchestration Yes -FinalReminderMinutesInterval 52 -  
InitialReminderHoursInterval 6 -InstallRestriction OnlyAdministrators -PortalUrl  
"http://NewInstall.Contoso.com" -PowerShellExecutionPolicy Bypass -SuspendBitLocker Always
```

Example 10: Configure restart settings

This command sets restart logoff notification countdown duration and logoff notification final window duration for a client setting object named TSQA11.

```
PS C:\> Set-CMClientSetting -Name "TSQA11" -RebootLogoffNotificationCountdownDuration 12 -  
RebootLogoffNotificationFinalWindowMinutes 80
```

Example 11: Configure metered network usage

This command specifies the type of metered network usage for the client setting object named TSQA21 as Limit.

```
PS C:\> Set-CMClientSetting -Name "TSQA21" -MeteredNetworkUsage Limit
```

Related topics

[Get-CMClientSetting](#)

[New-CMClientSetting](#)

[Remove-CMClientSetting](#)

[New-CMSchedule](#)

Set-CMClientStatusSetting

Set-CMClientStatusSetting

Modifies client status settings.

Syntax

Parameter Set: Set

```
Set-CMClientStatusSetting [-ClientPolicyDayInterval <Int32> ] [-HardwareInventoryDayInterval <Int32> ] [-HeartbeatDiscoveryDayInterval <Int32> ] [-HistoryCleanupDayInterval <Int32> ] [-SoftwareInventoryDayInterval <Int32> ] [-StatusMessageDayInterval <Int32> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Set-CMClientStatusSetting** cmdlet modifies client status settings. These settings determine the data collection intervals for individual client monitoring activities in Microsoft System Center 2012 Configuration Manager.

For more information about client settings, see [About Client Settings in Configuration Manager](http://go.microsoft.com/fwlink/?LinkId=266226) (<http://go.microsoft.com/fwlink/?LinkId=266226>) in the TechNet library.

Parameters

-ClientPolicyDayInterval<Int32>

Specifies the number of days since a client requested policy. The default value is 7 days.

Aliases	PolicyInactiveInterval
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-HardwareInventoryDayInterval<Int32>

Specifies the number of days since a client computer has sent a hardware inventory record to the site database. The default value is 7 days.

Aliases	HWInactiveInterval
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-HeartbeatDiscoveryDayInterval<Int32>

Specifies the number of days since a client computer sent a heartbeat discovery record to the site database. The default value is 7 days.

Aliases	DDRInactiveInterval
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-HistoryCleanupDayInterval<Int32>

Specifies how long you want the client status history to remain in the site database. The default value is 31 days.

Aliases	CleanUpInterval
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SoftwareInventoryDayInterval<Int32>

Specifies the number of days since a client computer sent a software inventory record to the site database. The default value is 7 days.

Aliases	SWInactiveInterval
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-StatusMessageDayInterval<Int32>

Specifies the number of days since a client computer sent status messages to the site database. The default value is 7 days.

Aliases	StatusInactiveInterval
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Modify all client status settings

This command modifies all client status settings.

```
PS C:\> Set-CMClientStatusSetting -ClientPolicyDayInterval 2 -HeartbeatDiscoveryDayInterval 3 -HardwareInventoryDayInterval 4 -SoftwareInventoryDayInterval 5 -StatusMessageDayInterval 6 -HistoryCleanupDayInterval 7
```

Example 2: Modify the Client Policy setting

This command modifies the client policy day setting only.

```
PS C:\> Set-CMClientStatusSetting -ClientPolicyDayInterval 5
```

Related topics

[Get-CMClientStatusSetting](#)

[Update-CMClientStatus](#)

[Get-CMClientStatusUpdateSchedule](#)

Set-CMClientStatusUpdateSchedule

Set-CMClientStatusUpdateSchedule

Modifies the schedule interval of the client status update task.

Syntax

Parameter Set: Set

```
Set-CMClientStatusUpdateSchedule -Interval <Int32> -UnitType {Days | Hours} [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Set-CMClientStatusUpdateSchedule** cmdlet modifies the schedule interval of the client status update task. For more information, see [How to Configure Client Status in Configuration Manager](http://go.microsoft.com/fwlink/?LinkId=247263) (<http://go.microsoft.com/fwlink/?LinkId=247263>) on TechNet.

Parameters

-Interval<Int32>

Specifies the number of hours or days between client status updates.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UnitType<ClientStatusUpdateScheduleUnit>

Specifies whether the interval between schedule updates is calculated in hours or days. Valid values are: Hours and Days.

The acceptable values for this parameter are:

Days	
Hours	

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Modify a client status update schedule

This command modifies the client status update schedule.

```
PS C:\> Set-CMClientStatusUpdateSchedule -Interval 23 -UnitType Hours
```

Related topics

[Get-CMClientStatusUpdateSchedule](#)

[Get-CMClientStatusSetting](#)

[Set-CMClientStatusSetting](#)

Set-CMCloudDistributionPoint

Set-CMCloudDistributionPoint

Changes settings for a cloud-based distribution point.

Syntax

Parameter Set: SetById

```
Set-CMCloudDistributionPoint -Id <String[]> [-Description <String> ] [-NewName <String> ] [-StorageQuotaGB <Int32> ] [-StorageQuotaGrow <Boolean> ] [-TrafficOutGB <Int32> ] [-TrafficOutStopService <Boolean> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetByName

```
Set-CMCloudDistributionPoint -Name <String> [-Description <String> ] [-NewName <String> ] [-StorageQuotaGB <Int32> ] [-StorageQuotaGrow <Boolean> ] [-TrafficOutGB <Int32> ] [-TrafficOutStopService <Boolean> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetByValueMandatory

```
Set-CMCloudDistributionPoint -InputObject <IResultObject> [-Description <String> ] [-NewName <String> ] [-StorageQuotaGB <Int32> ] [-StorageQuotaGrow <Boolean> ] [-TrafficOutGB <Int32> ] [-TrafficOutStopService <Boolean> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Set-CMCloudDistributionPoint** cmdlet changes settings for a cloud-based distribution point.

In Microsoft System Center 2012 Configuration Manager, you can use a cloud service in Windows Azure to host a distribution point for storing files to download to clients. You can send packages and apps to and host packages and apps in cloud distribution points. For more information about cloud distribution points, see [Planning for Content Management in Configuration Manager](http://go.microsoft.com/fwlink/?LinkId=266223) (<http://go.microsoft.com/fwlink/?LinkId=266223>) on TechNet.

You can use the **Set-CMCloudDistributionPoint** cmdlet to specify storage alert thresholds and warning levels for content that you deploy to a cloud distribution point. You can also use the cmdlet to configure settings that enable users and devices to access the content. You can provide a name and description for the cloud distribution point.

Parameters

-Description<String>

Specifies a description for a cloud distribution point.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Id<String[]>

Specifies an array of identifiers for one or more cloud distribution points. You can use a comma-separated list.

Aliases	AzureServiceId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a cloud distribution point object. To obtain a cloud distribution point object, you can use the **Get-CMCloudDistributionPoint** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies a name for a cloud distribution point.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-NewName<String>

Specifies a new name for the cloud-based distribution point.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-StorageQuotaGB<Int32>

Specifies the threshold value, in gigabytes, that triggers errors or warnings for total content storage.

Aliases	StorageQuotaInGB
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-StorageQuotaGrow<Boolean>

Specifies whether the storage quota can grow. By default, the amount of stored data cannot exceed the value of the *StorageQuotaInGB* parameter. The default value for this parameter is *\$False*.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-TrafficOutGB<Int32>

Specifies the threshold value, in gigabytes, that triggers errors or warnings, for monthly traffic out of Windows Azure Storage Service.

Aliases	TrafficOutInGB
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-TrafficOutStopService<Boolean>

Specifies whether Configuration Manager stops data transfers after the distribution point reaches the quota specified in the *TrafficOutInGB* parameter. The default value for this parameter is *\$False*.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Set values for a distribution point

This command sets the description and name for a distribution point to the provided strings. It also sets values for the storage quota and data transfer.

```
PS C:\> Set-CMCloudDistributionPoint -Id 16777237 -Description "Western distribution point"
-Name "West01" -StorageQuotaInGB 50 -TrafficOutInGB 50
```

Related topics

[Get-CMCloudDistributionPoint](#)

[New-CMCloudDistributionPoint](#)

[Remove-CMCloudDistributionPoint](#)

[Start-CMCloudDistributionPoint](#)

[Stop-CMCloudDistributionPoint](#)

Set- CMCollectionMembershipEvaluationComponent

Set-CMCollectionMembershipEvaluationComponent

Sets how often Configuration Manager evaluates collections for membership.

Syntax

Parameter Set: SearchByNameMandatory

```
Set-CMCollectionMembershipEvaluationComponent -MinutesInterval <Int32> -SiteSystemName  
<String[]> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchBySiteCodeMandatory

```
Set-CMCollectionMembershipEvaluationComponent -MinutesInterval <Int32> -SiteCode <String> [-  
SiteSystemName <String[]> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Set-CMCollectionMembershipEvaluationComponent** cmdlet changes how often Microsoft System Center 2012 Configuration Manager evaluates collections. System Center 2012 Configuration Manager queries the database at a regular interval to check for changes in collection membership. You can specify which site to change by site server name or site code.

Parameters

-MinutesInterval<Int32>

Specifies an evaluation interval, in minutes. Configuration Manager checks the members of a collection to verify that the members still belong in that collection.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-SiteCode<String>

Specifies an array of a site codes for Configuration Manager sites.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteSystemName<String[]>

Specifies an array of names for Configuration Manager servers.

Aliases	Name
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Set an evaluation period for a site code

This command sets the evaluation frequency to five minutes for the specified site code.

```
PS C:\> Set-CMCollectionMembershipEvaluationComponent -MinutesInterval 5 -SiteCode "CM4"
```

Example 2: Set an evaluation period for a system

This command sets the evaluation frequency to six minutes for the server named CM01.West01.Contoso.com.

```
PS C:\> Set-CMCollectionMembershipEvaluationComponent -MinutesInterval 6 -SiteSystemName "CM01.West01.Contoso.com"
```

Related topics

[Get-CMCollectionMembershipEvaluationComponent](#)

Set-CMComputerAssociation

Set-CMComputerAssociation

Changes settings for a computer association in Configuration Manager.

Syntax

Parameter Set: NewComputerAssociation

```
Set-CMComputerAssociation -DestinationComputer <String> -SourceComputer <String> [-  
AddMigrationUserName <String[]> ] [-MigrationBehavior  
{CaptureAllUserAccountsAndRestoreSpecifiedAccounts | CaptureAndRestoreAllUserAccounts |  
CaptureAndRestoreSpecifiedUserAccounts} ] [-RemoveMigrationUserName <String[]> ] [-Confirm]  
[-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Set-CMComputerAssociation** cmdlet changes settings for a computer association used for migration. Microsoft System Center 2012 Configuration Manager can migrate user state and settings from an existing computer to a different computer as part of operating system deployment. In the course of migration, System Center 2012 Configuration Manager saves accounts created on the source computer and creates those user accounts on the destination computer.

A computer association contains the user names to be migrated and how to deal with other user names from the source computer. You can use this cmdlet to modify an association. You can add user names to the association, or remove user names. You can also change whether System Center 2012 Configuration Manager includes other user names from the source computer.

Parameters

-AddMigrationUserName<String[]>

Specifies an array of user names for accounts created on the source computer. The cmdlet adds these user names to the current specified user names of the computer association.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DestinationComputer<String>

Specifies the name of a destination computer.

Aliases	RestoreName
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-MigrationBehavior<MigrationBehavior>

Specifies how Configuration Manager treats user accounts created on the source computer. When you create a computer association, specify user accounts created on the source computer by using the *MigrationUserName* parameter of the **New-CMComputerAssociation** cmdlet. The computer association can specify that the migration process creates some or all of those accounts on the destination computer.

Valid values are:

- CaptureAllUserAccountsAndRestoreSpecifiedAccounts. Saves all accounts created on the source computer, but creates only the specified accounts on the destination computer.
- CaptureAndRestoreAllUserAccounts. Saves all accounts created on the source computer, and creates them on the destination computer.
- CaptureAndRestoreSpecifiedUserAccounts. Saves only the specified accounts from the source computer, and creates those accounts on the destination computer.

The acceptable values for this parameter are:

CaptureAllUserAccountsAndRestoreSpecifiedAccounts	
CaptureAndRestoreAllUserAccounts	
CaptureAndRestoreSpecifiedUserAccounts	

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RemoveMigrationUserName<String[]>

Specifies an array of user names for accounts created on the source computer. The cmdlet removes these user names from current specified user names of the computer association.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SourceComputer<String>

Specifies the name of the source computer.

Aliases	SourceName
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Modify a computer association

This command changes the association between the computer named TSQA073 and TSQA155. The command adds the user ContosoTSQA\EvanNarvaez and removes the user ContosoTSQA\ElisaDaugherty. The command specifies the migration behavior as CaptureAllUserAccountsAndRestoreSpecifiedAccounts, so the association causes the migration to save all accounts created on the source computer, but only to create the accounts specified by the computer association on the destination computer.

```
PS C:\> Set-CMComputerAssociation -DestinationComputer "TSQA155" -SourceComputer "TSQA073" -  
AddMigrationUserName "ContosoTSQA\EvanNarvaez" -MigrationBehavior  
CaptureAllUserAccountsAndRestoreSpecifiedAccounts -RemoveMigrationUserName  
"ContosoTSQA\ElisaDaugherty"
```

Related topics

[Get-CMComputerAssociation](#)

[New-CMComputerAssociation](#)

[Remove-CMComputerAssociation](#)

Set-CMConfigurationItem

Set-CMConfigurationItem

Changes settings for a Configuration Manager configuration item.

Syntax

Parameter Set: SetByIdMandatory

```
Set-CMConfigurationItem -Id <String[]> [-AddCategory <String[]> ] [-Description <String> ]  
[-DesiredConfigurationDigestPath <String> ] [-NewName <String> ] [-RemoveCategory <String[]>  
] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetByNameMandatory

```
Set-CMConfigurationItem -Name <String[]> [-AddCategory <String[]> ] [-Description <String> ]  
[-DesiredConfigurationDigestPath <String> ] [-NewName <String> ] [-RemoveCategory <String[]>  
] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetByValueMandatory

```
Set-CMConfigurationItem -InputObject <IResultObject> [-AddCategory <String[]> ] [-  
Description <String> ] [-DesiredConfigurationDigestPath <String> ] [-NewName <String> ] [-  
RemoveCategory <String[]> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetSecurityScopeById

```
Set-CMConfigurationItem -Id <String[]> -SecurityScopeAction {AddMembership |  
RemoveMembership} -SecurityScopeName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetSecurityScopeByName

```
Set-CMConfigurationItem -Name <String[]> -SecurityScopeAction {AddMembership |  
RemoveMembership} -SecurityScopeName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetSecurityScopeByValue

```
Set-CMConfigurationItem -InputObject <IResultObject> -SecurityScopeAction {AddMembership |  
RemoveMembership} -SecurityScopeName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Set-CMConfigurationItem** cmdlet changes settings for a Microsoft System Center 2012 Configuration Manager configuration item.

Configuration items contain one or more settings, along with compliance rules. Items usually define a unit of configuration you want to monitor. For more information about configuration items, see

[Introduction to Compliance Settings in Configuration Manager](http://go.microsoft.com/fwlink/?LinkId=211014)

(<http://go.microsoft.com/fwlink/?LinkId=211014>) on TechNet.

Parameters

-AddCategory<String[]>

Specifies an array of localized names of the categories to which the configuration item belongs.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Description<String>

Specifies a description for a configuration item.

Aliases	LocalizedDescription
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DesiredConfigurationDigestPath<String>

Specifies a path and file name for the digest file of this configuration item.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Id<String[]>

Specifies an array of identifiers for one or more configuration items. You can use a comma separated list.

Aliases	CIId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a configuration item object. To obtain a configuration item object, you can use the **Get-CMConfigurationItem** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String[]>

Specifies an array of names for configuration items.

Aliases	LocalizedDisplayName
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-NewName<String>

Specifies a new name for a configuration item.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RemoveCategory<String[]>

Specifies an array of localized names of the categories from which to remove the configuration item.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SecurityScopeAction<SecurityScopeActionType>

Specifies an action for a security scope. Valid values are: AddMembership and RemoveMembership.

The acceptable values for this parameter are:

AddMembership	
RemoveMembership	

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SecurityScopeName<String>

Specifies the name of a security scope. A security scope name can be Default or the name of a custom security scope.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Change the name of a configuration item

This command changes the name of the item named CITest to CITest01.

```
PS C:\> Set-CMConfigurationItem -Name "CITest" -NewName "CITest01"
```

Example 2: Set item settings

This command sets the security scope action to AddMembership and the security scope name to DefaultScope for the item named CITest01.

```
PS C:\> Set-CMConfigurationItem -Name "CITest01" -SecurityScopeAction AddMembership -  
SecurityScopeName "DefaultScope"
```

Example 3: Change item settings

This command sets the security scope action to RemoveMembership and the security scope name to DefaultScope for the item named CITest01.

```
PS C:\> Set-CMConfigurationItem -Name "CITest01" -SecurityScopeAction RemoveMembership -  
SecurityScopeName "DefaultScope"
```

Related topics

[Export-CMConfigurationItem](#)

[Get-CMConfigurationItem](#)

[Get-CMConfigurationItemXMLDefinition](#)

[Import-CMConfigurationItem](#)

[New-CMConfigurationItem](#)

[Remove-CMConfigurationItem](#)

[Get-CMConfigurationItemHistory](#)

Set-CMConfigurationPolicyDeployment

Set-CMConfigurationPolicyDeployment

Creates a configuration policy deployment.

Syntax

Parameter Set: SetFWPolicyDeploymentByIdMandatory

```
Set-CMConfigurationPolicyDeployment -CollectionName <String> -FirewallPolicyId <String> [-Schedule <IResultObject> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetFWPolicyDeploymentByNameMandatory

```
Set-CMConfigurationPolicyDeployment -CollectionName <String> -FirewallPolicyName <String> [-Schedule <IResultObject> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetFWPolicyDeploymentByValueMandatory

```
Set-CMConfigurationPolicyDeployment -CollectionName <String> -FirewallPolicy <IResultObject> [-Schedule <IResultObject> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetUSMPolicyDeploymentByIdMandatory

```
Set-CMConfigurationPolicyDeployment -CollectionName <String> -UserDataAndProfileId <String> [-EnableEnforcement <Boolean> ] [-GenerateAlert <Boolean> ] [-MonitoredByScom <Boolean> ] [-OverrideServiceWindow <Boolean> ] [-ParameterValue <Int32> ] [-PostponeDate <DateTime> ] [-PostponeTime <DateTime> ] [-Schedule <IResultObject> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetUSMPolicyDeploymentByNameMandatory

```
Set-CMConfigurationPolicyDeployment -CollectionName <String> -UserDataAndProfileName <String> [-EnableEnforcement <Boolean> ] [-GenerateAlert <Boolean> ] [-MonitoredByScom <Boolean> ] [-OverrideServiceWindow <Boolean> ] [-ParameterValue <Int32> ] [-PostponeDate <DateTime> ] [-PostponeTime <DateTime> ] [-Schedule <IResultObject> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetUSMPolicyDeploymentByValueMandatory

```
Set-CMConfigurationPolicyDeployment -CollectionName <String> -UserDataAndProfile <IResultObject> [-EnableEnforcement <Boolean> ] [-GenerateAlert <Boolean> ] [-MonitoredByScom <Boolean> ] [-OverrideServiceWindow <Boolean> ] [-ParameterValue <Int32> ] [-PostponeDate <DateTime> ] [-PostponeTime <DateTime> ] [-Schedule <IResultObject> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Set-CMConfigurationPolicyDeployment** cmdlet creates a configuration policy deployment in Microsoft System Center 2012 Configuration Manager. You can deploy firewall policies or user session management policies. Use the [Start-CMConfigurationPolicyDeployment](#) cmdlet to deploy specified policies for a System Center 2012 Configuration Manager collection.

Parameters

-CollectionName<String>

Specifies the name of a collection. The deployment applies to this collection.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnableEnforcement<Boolean>

Specifies whether to enable enforcement for the deployment. During enforcement, a client reports compliance information about a deployment.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-FirewallPolicy<IResultObject>

Specifies a Windows Firewall Policy object. To obtain a **CMWindowsFirewallPolicy** object, use the [Get-CMWindowsFirewallPolicy](#) cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-FirewallPolicyId<String>

Specifies the ID of a Windows Firewall policy.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-FirewallPolicyName<String>

Specifies the name of a Windows Firewall policy.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-GenerateAlert<Boolean>

Indicates whether Configuration Manager generates alerts during the deployment.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-MonitoredByScom<Boolean>

Specifies whether System Center 2012 – Operations Manager monitoring criteria applies during the deployment.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OverrideServiceWindow<Boolean>

Indicates whether to override the service window while deploying policies. Service windows are periods of time allocated for maintenance.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ParameterValue<Int32>

Specifies the values of administrator-defined parameters, such as thresholds. Configuration Manager stores the values in XML format.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PostponeDate<DateTime>

Specifies a date, as a **DateTime** object, for the deployment if it is postponed. To obtain a **DateTime** object, use the **Get-Date** cmdlet. For more information, type `Get-Help Get-Date`.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PostponeTime<DateTime>

Specifies a time, as a **DateTime** object, for the deployment if it is postponed. To obtain a **DateTime** object, use the **Get-Date** cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Schedule<IResultObject>

Specifies a schedule object. This is the schedule for deploying the configuration policy. You can use the [New-CMSchedule](#) cmdlet to create a schedule token.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UserDataAndProfile<IResultObject>

Specifies a user data and profiles configuration item object. To obtain a **CMUserDataAndProfileConfigurationItem** object, use the [Get-CMUserDataAndProfileConfigurationItem](#) cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UserDataAndProfileId<String>

Specifies an ID of a user data and profile configuration item.

Aliases	none
Required?	true
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UserDataAndProfileName<String>

Specifies a name of a user data and profile configuration item.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Create a configuration policy deployment

This command creates a configuration policy deployment named Remote Firewall Policy and deploys it to the collection named Regional Remote Users.

```
PS C:\> Set-CMConfigurationPolicyDeployment -CollectionName "Regional Remote Users" -  
FirewallPolicyName "Remote Firewall Policy"
```

Related topics

[Start-CMConfigurationPolicyDeployment](#)

[Get-CMUserDataAndProfileConfigurationItem](#)

[New-CMSchedule](#)

[Get-CMWindowsFirewallPolicy](#)

Set-CMDatabaseProperty

Set-CMDatabaseProperty

Changes database settings for a Configuration Manager database.

Syntax

Parameter Set: SearchBySiteCode

```
Set-CMDatabaseProperty -SiteCode <String> [-DataRetentionPeriodDays <Int32> ] [-  
EnableDataCompression <Boolean> ] [-SqlServerServiceBrokerPort <Int32> ] [-Confirm] [-  
WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Set-CMDatabaseProperty** cmdlet changes database settings for a Microsoft System Center 2012 Configuration Manager site database. Specify the System Center 2012 Configuration Manager site code for the database that you want to modify.

You can modify whether the database uses data compression, the Service Broker port for the computer that runs Microsoft SQL Server, and the length of time that the database keeps data. You can use the **Get-CMDatabaseProperty** cmdlet to see current values for these properties.

Parameters

-DataRetentionPeriodDays<Int32>

Specifies a length of time, in days, that the database stores data. After data exceeds this period, the database removes it during the next scheduled grooming.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnableDataCompression<Boolean>

Specifies whether the database uses data compression.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteCode<String>

Specifies the site code for a Configuration Manager site.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SqlServerServiceBrokerPort<Int32>

Specifies the port that the computer running SQL Server uses as a Service Broker port.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Change settings for a database

This command makes changes to the database for the site that has the site code CM2. The command sets the data retention period to 10 days, disables data compression, and specifies a port for the SQL Server Service Broker.

```
PS C:\> Set-CMDatabaseProperty -SiteCode "CM2" -DataRetentionPeriodDays 10 -  
EnableDataCompression $False -SqlServerServiceBrokerPort 80
```

Related topics

[Get-CMDatabaseProperty](#)

Set-CMDatabaseReplicationLinkProperty

Set-CMDatabaseReplicationLinkProperty

Changes configuration settings for a database replication link.

Syntax

Parameter Set: SetBySiteCodeMandatory

```
Set-CMDatabaseReplicationLinkProperty -ChildSiteCode <String> -ParentSiteCode <String> [-  
DegradedLinkStatusRetryCount <Int32> ] [-EnableDistributedViewForHardwareInventory <Boolean>  
] [-EnableDistributedViewForSoftwareInventory <Boolean> ] [-  
EnableDistributedViewForStatusMessage <Boolean> ] [-FailedLinkStatusRetryCount <Int32> ] [-  
GenerateReplicationDownAlert <Boolean> ] [-  
ReplicationDataTrafficSummarizationIntervalMinutes <Int32> ] [-  
ReplicationDownAlertThresholdMinutes <Int32> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetScheduleBySiteCodeMandatory

```
Set-CMDatabaseReplicationLinkProperty -AvailabilityLevel {Closed | HINV | SINV | HINVSINV |  
StatMSG | HINVStatMSG | SINVStatMSG | HINVSINVStatMSG} -ChildSiteCode <String> -DaysOfWeek  
{Friday | Monday | Saturday | Sunday | Thursday | Tuesday | Wednesday} -ParentSiteCode  
<String> -TimePeriodEnd <Int32> -TimePeriodStart <Int32> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Detailed Description

The **Set-CMDatabaseReplicationLinkProperty** cmdlet changes configuration settings for a database replication link between a Microsoft System Center 2012 Configuration Manager parent site and child site.

Database replication for System Center 2012 Configuration Manager sites transfers data and merges changes made in a site database with information stored at other sites in the System Center 2012 Configuration Manager hierarchy. This enables all sites to share the same information.

Parameters

-AvailabilityLevel<InvAvailabilityLevel>

Specifies the availability level for software and hardware inventory on a client computer. Valid values are:

- Closed
- HINV

- SINV
- HINVSINV
- StatMSG
- HINVStatMSG
- SINVStatMSG
- HINVSINVStatMSG

The acceptable values for this parameter are:

Closed	
HINV	
SINV	
HINVSINV	
StatMSG	
HINVStatMSG	
SINVStatMSG	
HINVSINVStatMSG	

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ChildSiteCode<String>

Specifies a site code for a Configuration Manager site. This parameter refers to the child site.

Aliases	Site2
Required?	true
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DaysOfWeek<DaysOfWeek[]>

Specifies an array of day names that determine the days of each week on which Configuration Manager replicates the database for Configuration Manager sites. Valid values are:

- Monday
- Tuesday
- Wednesday
- Thursday
- Friday
- Saturday
- Sunday

The acceptable values for this parameter are:

Friday	
Monday	
Saturday	
Sunday	
Thursday	
Tuesday	
Wednesday	

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DegradedLinkStatusRetryCount<Int32>

Specifies a retry count when a replication group or object is delayed due to degraded link status.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnableDistributedViewForHardwareInventory<Boolean>

Indicates whether Configuration Manager configures the SQL Server distributed partitioned views for hardware inventory.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnableDistributedViewForSoftwareInventory<Boolean>

Indicates whether Configuration Manager configures the SQL Server distributed partitioned views for software inventory.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnableDistributedViewForStatusMessage<Boolean>

Indicates whether Configuration Manager configures the SQL Server distributed partitioned views for status messages.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-FailedLinkStatusRetryCount<Int32>

Specifies a retry count when a replication group or object is delayed by failed link status.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-GenerateReplicationDownAlert<Boolean>

Indicates whether to generate a replication down alert.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-ParentSiteCode<String>

Specifies a site code for a Configuration Manager site. This parameter refers to the parent site.

Aliases	Site1
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ReplicationDataTrafficSummarizationIntervalMinutes<Int32>

Specifies the interval, in minutes, at which Configuration Manager summarizes report data for database replication traffic for the link.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ReplicationDownAlertThresholdMinutes<Int32>

Specifies a threshold, in minutes, to elapse before Configuration Manager issues a replication down alert. Use this threshold to avoid sending alerts in response to false warnings.

Aliases	none
Required?	false
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-TimePeriodEnd<Int32>

Specifies an end time, in hours, of the period of time to wait before replication failures trigger the site to reinitializes its copy of the site database.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-TimePeriodStart<Int32>

Specifies a start time, in hours, of the period of time to wait before replication failures trigger the site to reinitializes its copy of the site database.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Change settings of a database replication link

This command changes configuration settings for a database replication link between the Configuration Manager parent site that has the site code CCC and the child site that has the site code CCB.

```
PS C:\> Set-CMDatabaseReplicationLinkProperty -ParentSiteCode "CCC" -ChildSiteCode "CCB" -
EnableDistributedViewForHardwareInventory 1 -EnableDistributedViewForSoftwareInventory 1 -
EnableDistributedViewForStatusMessage 1 -ReplicationDataTrafficSummarizationIntervalMinutes
12 -DegradedLinkStatusRetryCount 40 -FailedLinkStatusRetryCount 60 -
GenerateReplicationDownAlert 1 -ReplicationDownAlertThresholdMinutes 20
```

Example 2: Set the schedule for a database replication link

This command sets the schedule for the database replication link between the Configuration Manager parent site that has the site code CCC and the child site that has the site code CCB. The command specifies that Configuration Manager replicates the database for Configuration Manager sites on Friday,

Monday and Tuesday. The command specifies software and hardware inventory availability on the client computer.

```
PS C:\> Set-CMDatabaseReplicationLinkProperty -ParentSiteCode "CCC" -ChildSiteCode "CCB" -  
DaysOfWeek Friday, Monday, Tuesday -TimePeriodStart 8 -TimePeriodEnd 0 -AvailabilityLevel  
HINVSINV
```

Related topics

[Get-CMDatabaseReplicationLinkProperty](#)

[Get-CMDataBaseReplicationStatus](#)

Set-CMDeploymentType

Set-CMDeploymentType

Changes a deployment type.

Syntax

Parameter Set: SetByIdPriority

```
Set-CMDeploymentType -ApplicationName <String> -DeploymentTypeId <String> [-Priority {Decrease | Increase} ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetByNamePriority

```
Set-CMDeploymentType -ApplicationName <String> -DeploymentTypeName <String> [-Priority {Decrease | Increase} ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetByNamePropertyAppV5xInstaller

```
Set-CMDeploymentType -ApplicationName <String> -AppV5xInstaller -DeploymentTypeName <String> [-AdministratorComment <String> ] [-AllowClientsToUseFallbackSourceLocationForContent <Boolean> ] [-ContentLocation <String> ] [-EnablePeertoPeerContentDistribution <Boolean> ] [-Language <String[]> ] [-NewDeploymentTypeName <String> ] [-OnFastNetworkMode {RunFromNetwork | RunLocal} ] [-OnSlowNetworkMode {DoNothing | Download | DownloadContentForStreaming} ] [-PersistContentInClientCache <Boolean> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetByNamePropertyAppVInstaller

```
Set-CMDeploymentType -ApplicationName <String> -AppVInstaller -DeploymentTypeName <String> [-AdministratorComment <String> ] [-AllowClientsToUseFallbackSourceLocationForContent <Boolean> ] [-ContentLocation <String> ] [-EnablePeertoPeerContentDistribution <Boolean> ] [-Language <String[]> ] [-LoadContentIntoAppVcacheBeforelaunch <Boolean> ] [-NewDeploymentTypeName <String> ] [-OnFastNetworkMode {RunFromNetwork | RunLocal} ] [-OnSlowNetworkMode {DoNothing | Download | DownloadContentForStreaming} ] [-PersistContentInClientCache <Boolean> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetByNamePropertyDeepLinkInstaller

```
Set-CMDeploymentType -ApplicationName <String> -DeepLinkInstaller -DeploymentTypeName <String> [-AdministratorComment <String> ] [-ApplicationNameInWindowsStore <String> ] [-Language <String[]> ] [-NewDeploymentTypeName <String> ] [-RemoteComputerName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetByNamePropertyMacInstaller

```
Set-CMDeploymentType -ApplicationName <String> -DeploymentTypeName <String> -MacInstaller [-AdministratorComment <String> ] [-ContentLocation <String> ] [-InstallationProgram <String> ] [-Language <String[]> ] [-MacRebootBehavior {ForceReboot | NoAction} ] [-NewDeploymentTypeName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetByNamePropertyMsiConfigureRule

```
Set-CMDeploymentType -ApplicationName <String> -DeploymentTypeName <String> -
```

MsiOrScriptInstaller [-AdministratorComment <String>] [-
AllowClientsToShareContentOnSameSubnet <Boolean>] [-
AllowClientsToUseFallbackSourceLocationForContent <Boolean>] [-ContentLocation <String>]
[-DetectDeploymentTypeByCustomScript] [-EstimatedInstallationTimeMinutes <Int32>] [-
InstallationBehaviorType {InstallForSystem |
InstallForSystemIfResourceIsDeviceOtherwiseInstallForUser | InstallForUser}] [-
InstallationProgram <String>] [-InstallationProgramVisibility {Normal | Minimized |
Maximized | Hidden}] [-InstallationStartIn <String>] [-Language <String[]>] [-
LogonRequirementType {OnlyWhenNoUserLoggedIn | OnlyWhenUserLoggedIn | WhereOrNotUserLoggedIn
| WhetherOrNotUserLoggedIn}] [-MaximumAllowedRunTimeMinutes <Int32>] [-
NewDeploymentTypeName <String>] [-OnSlowNetworkMode {DoNothing | Download |
DownloadContentForStreaming}] [-PersistContentInClientCache <Boolean>] [-ProductCode
<String>] [-RebootBehavior {BasedOnExitCode | ForceReboot | NoAction | ProgramReboot}] [-
RequiresUserInteraction <Boolean>] [-
RunInstallationAndUninstallProgramAs32bitProcessOn64bitClient <Boolean>] [-
RunScriptAs32bitProcessOn64bitClient <Boolean>] [-ScriptContent <String>] [-ScriptType
<ScriptLanguage>] [-UninstallProgram <String>] [-UninstallStartIn <String>] [-Confirm] [-
WhatIf] [<CommonParameters>]

Parameter Set: SetByNamePropertyOtherInstaller

Set-CMDeploymentType -ApplicationName <String> -DeploymentTypeName <String> [-
AdministratorComment <String>] [-ContentLocation <String>] [-Language <String[]>] [-
NewDeploymentTypeName <String>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SetByNamePropertyWebAppInstaller

Set-CMDeploymentType -ApplicationName <String> -DeploymentTypeName <String> -WebAppInstaller
[-AdministratorComment <String>] [-Language <String[]>] [-NewDeploymentTypeName <String>]
[-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SetByNamePropertyWindows8Installer

Set-CMDeploymentType -ApplicationName <String> -DeploymentTypeName <String> -
Windows8AppInstaller [-AdministratorComment <String>] [-
AllowClientsToShareContentOnSameSubnet <Boolean>] [-
AllowClientsToUseFallbackSourceLocationForContent <Boolean>] [-ContentLocation <String>]
[-Language <String[]>] [-MaximumAllowedRunTimeMinutes <Int32>] [-NewDeploymentTypeName
<String>] [-OnSlowNetworkMode {DoNothing | Download | DownloadContentForStreaming}] [-
PersistContentInClientCache <Boolean>] [-TriggerVPN <Boolean>] [-Confirm] [-WhatIf] [
<CommonParameters>]

Parameter Set: SetByNamePropertyWmInstaller

Set-CMDeploymentType -ApplicationName <String> -DeploymentTypeName <String> -WmInstaller [-
AdministratorComment <String>] [-AllowUserToUninstall <Boolean>] [-ContentLocation
<String>] [-Language <String[]>] [-NewDeploymentTypeName <String>] [-Confirm] [-WhatIf] [
<CommonParameters>]

Parameter Set: SetByValuePriority

Set-CMDeploymentType -ApplicationName <String> -InputObject <IResultObject> [-Priority
{Decrease | Increase}] [-Confirm] [-WhatIf] [<CommonParameters>]

Detailed Description

The **Set-CMDeploymentType** cmdlet changes a deployment type for a deployment application in Microsoft System Center 2012 Configuration Manager. A deployment type is a part of the application that defines how that application deploys other applications to devices. You can also use this cmdlet to change the priority for dependencies of the deployment type. System Center 2012 Configuration Manager evaluates and installs dependencies of a deployment type in order of priorities before it installs the deployment type.

Parameters

-AdministratorComment<String>

Specifies a description for the deployment type.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AllowClientsToShareContentOnSameSubnet<Boolean>

Indicates whether clients can share content with other clients on the same subnet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AllowClientsToUseFallbackSourceLocationForContent<Boolean>

Indicates whether clients can use a fallback location provided by a management point. A fallback location point provides an alternate location for source content when the content for the deployment type is not available on any preferred distribution points.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AllowUserToUninstall<Boolean>

Indicates whether a user can uninstall an application.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ApplicationName<String>

Specifies the name of the deployment application that contains the deployment type.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ApplicationNameInWindowsStore<String>

Specifies the name of the application in the Windows Store.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AppV5xInstaller

Indicates that the deployment type detects application information and deployment types from a Microsoft Application Virtualization 5 (.appv) package file.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AppVInstaller

Indicates that the deployment type detects application information and deployment types from a Microsoft Application Virtualization 5 (.appv) package file.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ContentLocation<String>

Specifies the path of the content. The site system server requires permission to read the content files.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeepLinkInstaller

Indicates that the deployment type detects application information and deployment types by providing a link to the application (in the Windows Store) on a computer where the application is already installed.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeploymentTypeId<String>

Specifies the type ID for a deployment type.

Aliases	CId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeploymentTypeName<String>

Specifies the name of a deployment type.

Aliases	LocalizedDisplayName
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DetectDeploymentTypeByCustomScript

Indicates that the deployment type uses a custom script to detect the presence of this deployment type.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnablePeertoPeerContentDistribution<Boolean>

Indicates whether clients can distribute content to other clients.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EstimatedInstallationTimeMinutes<Int32>

Specifies, in minutes, the estimated installation time.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a deployment type object for Configuration Manager. To obtain a deployment type object, use the **Get-CMDeploymentType** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InstallationBehaviorType<InstallationBehaviorType>

Specifies the installation behavior of the deployment type. Valid values are:

- InstallForSystem
- InstallForSystemIfResourceIsDeviceOtherwiseInstallForUser
- InstallForUser

The acceptable values for this parameter are:

InstallForSystem	
InstallForSystemIfResourceIsDeviceOtherwiseInstallForUser	
InstallForUser	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InstallationProgram<String>

Specifies the command line for the Windows Installer.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InstallationProgramVisibility<UserInteractionMode>

Specifies the mode in which the deployment type runs on client devices. Valid values are:

- Normal
- Minimized
- Maximized
- Hidden

The acceptable values for this parameter are:

Normal	
Minimized	
Maximized	
Hidden	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InstallationStartIn<String>

Specifies the folder that contains the installation program for the deployment type. This folder can be an absolute path on the client, or a path to the distribution point folder that contains the installation files.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Language<String[]>

Specifies an array of languages that the deployment type supports.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-LoadContentIntoAppVcacheBeforelaunch<Boolean>

Indicates whether to load the content into the AppV cache when you deploy the application.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-LogonRequirementType<LogonRequirementType>

Specifies the logon requirement for the deployment type. Valid values are:

- OnlyWhenNoUserLoggedIn
- OnlyWhenUserLoggedIn
- WhereOrNotUserLoggedIn

The acceptable values for this parameter are:

OnlyWhenNoUserLoggedIn	
OnlyWhenUserLoggedIn	
WhereOrNotUserLoggedIn	
WhetherOrNotUserLoggedIn	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-MacInstaller

Indicates that the deployment type detects application information and deployment types from a Mac OS X Installer (.cmmac) file that was created by using the CMAAppUtil tool.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-MacRebootBehavior<MacRebootBehavior>

Specifies the reboot behavior for computers running Mac OS X software.

The acceptable values for this parameter are:

ForceReboot	
NoAction	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-MaximumAllowedRunTimeMinutes<Int32>

Specifies, in minutes, the maximum time that the program is expected to run on the client computer.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-MsiOrScriptInstaller

Indicates that the deployment uses a script installer program.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-NewDeploymentTypeName<String>

Specifies the name of a new deployment type.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OnFastNetworkMode<OnFastNetworkMode>

Specifies the installation behavior of the deployment type on a fast network. Valid values are:

-- RunFromNetwork

-- RunLocal

The acceptable values for this parameter are:

RunFromNetwork	
RunLocal	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OnSlowNetworkMode<ContentHandlingMode>

Specifies the installation behavior of the deployment type on a slow network. Valid values are:

- DoNothing
- Download
- DownloadContentForStreaming

The acceptable values for this parameter are:

DoNothing	
Download	
DownloadContentForStreaming	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PersistContentInClientCache<Boolean>

Indicates whether the deployment type saves content in cache indefinitely on the client computer.

Aliases	none
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Priority<PriorityChangeType>

Specifies a change for the priority of the deployment type. Valid values are: Increase and Decrease.

The acceptable values for this parameter are:

Decrease	
Increase	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ProductCode<String>

Specifies a product code.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RebootBehavior<RebootBehavior>

Specifies the reboot behavior of the client computer.

The acceptable values for this parameter are:

BasedOnExitCode	
ForceReboot	
NoAction	
ProgramReboot	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RemoteComputerName<String>

Specifies a remote computer name.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RequiresUserInteraction<Boolean>

Indicates whether a user can interact with the deployment type installation to configure the installation options.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-

RunInstallationAndUninstallProgramAs32bitProcessOn64bitClient<Boolean>

Indicates whether to run the install and uninstall programs as 32-bit processes on a 64-bit client computer.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RunScriptAs32bitProcessOn64bitClient<Boolean>

Indicates whether the deployment type uses Microsoft Windows-32-on-Windows-64 (WOW64) subsystem to run a script on a 64-bit client computer.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ScriptContent<String>

Specifies the script language that you want to use to detect the deployment type.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ScriptType<ScriptLanguage>

Specifies the script language that you want to use to detect the deployment type.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-TriggerVPN<Boolean>

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UninstallProgram<String>

Specifies the name of the uninstall program and any parameters it requires.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UninstallStartIn<String>

Specifies the folder that contains the uninstall program for the deployment type. This folder can be an absolute path on the client, or a path that is relative to the distribution point folder that contains the package.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-WebAppInstaller

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Windows8AppInstaller

Indicates that the deployment type detects application information and deployment types from a Windows app package (.appx) file.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-WMInstaller

Indicates that the deployment type detects application information and deployment types from a Windows Mobile cabinet (.cab) file.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Increase the priority of a deployment application

This command sets a deployment type named Configuration Manager Console - Windows Installer (Native) for a deployment application named 2 - Child and increases the priority of that application.

```
PS C:\> Set-CMDeploymentType -ApplicationName "2 - Child" -DeploymentTypeName "Configuration Manager Console - Windows Installer (Native)" -Priority Increase
```

Example 2: Decrease the priority of a deployment application

This command sets a deployment type named Configuration Manager Console - Windows Installer (Native) for a deployment application named 2 - Child and decreases the priority of that application.

```
PS C:\> Set-CMDeploymentType -ApplicationName "2 - Child" -DeploymentTypeName "Configuration Manager Console - Windows Installer (Native)" -Priority Decrease
```

Related topics

[Add-CMDeploymentType](#)

[Get-CMDeploymentType](#)

[Remove-CMDeploymentType](#)

Set-CMDeviceCollection

Set-CMDeviceCollection

Sets the properties of one or more device collections in the Configuration Manager hierarchy.

Syntax

Parameter Set: SetByIdMandatory

```
Set-CMDeviceCollection -CollectionId <String> [-Comment <String> ] [-LimitingCollectionId <String> ] [-LimitingCollectionName <String> ] [-NewName <String> ] [-Confirm] [-WhatIf] [<CommonParameters>]
```

Parameter Set: SetByNameMandatory

```
Set-CMDeviceCollection -Name <String> [-Comment <String> ] [-LimitingCollectionId <String> ] [-LimitingCollectionName <String> ] [-NewName <String> ] [-Confirm] [-WhatIf] [<CommonParameters>]
```

Parameter Set: SetValueMandatory

```
Set-CMDeviceCollection -InputObject <IResultObject> [-Comment <String> ] [-LimitingCollectionId <String> ] [-LimitingCollectionName <String> ] [-NewName <String> ] [-Confirm] [-WhatIf] [<CommonParameters>]
```

Detailed Description

The **Set-CMDeviceCollection** cmdlet updates the name, description, or limiting collection of device collections.

Collections represent logical groupings of resources, such as users and devices. For more information about Microsoft System Center 2012 Configuration Manager collections, see [Introduction to Collections in Configuration Manager](http://go.microsoft.com/fwlink/p/?LinkID=259433) (http://go.microsoft.com/fwlink/p/?LinkID=259433) on TechNet.

Parameters

-CollectionId<String>

Specifies the IDs of the device collections that you want to update.

Aliases	none
Required?	true

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Comment<String>

Specifies a description of the specified device collections.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies an object that represents the device collections that you want to update.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-LimitingCollectionId<String>

Specifies the ID of the limiting collection for the specified device collections.

Aliases	LimitToCollectionId
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-LimitingCollectionName<String>

Specifies the name of the limiting collection for the specified device collections.

Aliases	LimitToCollectionName
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies the names of the device collections that you want to update.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-NewName<String>

Specifies a name for the specified device collections.

Aliases	none
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Update a device collection

This command updates the device collection that has the ID 9990000D.

PS C:\> Set-CMDeviceCollection -CollectionId "9990000D"

Related topics

[Export-CMDeviceCollection](#)

[Get-CMDeviceCollection](#)

[Import-CMDeviceCollection](#)

[New-CMDeviceCollection](#)

[Remove-CMDeviceCollection](#)

Set-CMDeviceCollectionVariable

Set-CMDeviceCollectionVariable

Sets the properties of a task sequence variable of a device collection in the Configuration Manager hierarchy.

Syntax

Parameter Set: SetByNameMandatory

```
Set-CMDeviceCollectionVariable -CollectionName <String> -VariableName <String> [-IsMask <Boolean> ] [-NewVariableName <String> ] [-NewVariableValue <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetByIdMandatory

```
Set-CMDeviceCollectionVariable -CollectionId <String> -VariableName <String> [-IsMask <Boolean> ] [-NewVariableName <String> ] [-NewVariableValue <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetByValueMandatory

```
Set-CMDeviceCollectionVariable -Collection <IResultObject> -VariableName <String> [-IsMask <Boolean> ] [-NewVariableName <String> ] [-NewVariableValue <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Set-CMDeviceCollectionVariable** cmdlet updates the name and value of a task sequence variable.

Task sequence variables are a set of name and value pairs that provide a mechanism to configure and customize the steps of a task sequence when the task sequence is deployed to a specific collection.

For more information about task sequence variables in Microsoft System Center 2012

Configuration Manager, see [Planning a Task Sequence Strategy in Configuration Manager](http://go.microsoft.com/fwlink/p/?LinkID=260806)

(<http://go.microsoft.com/fwlink/p/?LinkID=260806>) on TechNet.

Parameters

-Collection<IResultObject>

Specifies an object that represents the device collection. To obtain a device collection object, use the **Get-CMDeviceCollection** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CollectionId<String>

Specifies the ID of a device collection.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CollectionName<String>

Specifies the name of the device collection.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-IsMask<Boolean>

Indicates whether a value displays in the Configuration Manager console.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-NewVariableName<String>

Specifies a name for the variable.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-NewVariableValue<String>

Specifies a value for the variable

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-VariableName<String>

Specifies the name of the variable to update.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Set a task sequence variable for a device collection

This command sets the task sequence variable named New_ComputerName for the device collection named All Unknown Devices.

```
PS C:\> Set-CMDeviceCollectionVariable -CollectionName "All Unknown Devices" -VariableName "New_ComputerName"
```

Related topics

[Get-CMDeviceCollectionVariable](#)

[New-CMDeviceCollectionVariable](#)

[Remove-CMDeviceCollectionVariable](#)

[Get-CMDeviceCollection](#)

Set-CMDeviceOwnership

Set-CMDeviceOwnership

Configures ownership type for a device.

Syntax

Parameter Set: SearchByNameMandatory

```
Set-CMDeviceOwnership -DeviceName <String[]> -OwnershipType {Company | Personal} [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Set-CMDeviceOwnership -DeviceId <String[]> -OwnershipType {Company | Personal} [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Set-CMDeviceOwnership -InputObject <IResultObject> -OwnershipType {Company | Personal} [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Set-CMDeviceOwnership** cmdlet configures ownership type for a modern device. For a personal device, the information gathered is limited, and personal information is not removed during a wipe operation. For a company-owned device, additional information can be gathered and deleted during a wipe operation.

Parameters

-DeviceId<String[]>

Specifies an array of device IDs.

Aliases	ResourceId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-DeviceName<String[]>

Specifies an array of device names.

Aliases	Name
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a **CMDevice** object. To obtain a **CMDevice** object, use the **Get-CMDevice** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OwnershipType<DeviceOwnershipType>

Specifies the type of ownership for a device. Valid values are:

- Company. The device is a company asset.
- Personal. The device is not a company asset.

The acceptable values for this parameter are:

Company	
Personal	

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Identify a device as a company asset

This command identifies the specified device as a company asset.

```
PS C:\> Set-CMDeviceOwnership -DeviceId "209846738" -OwnershipType Company
```

Related topics

[Invoke-CMDeviceWipe](#)

[Get-CMDevice](#)

Set-CMDeviceVariable

Set-CMDeviceVariable

Modifies a device variable.

Syntax

Parameter Set: SearchByNameMandatory

```
Set-CMDeviceVariable -DeviceName <String> -IsMask <Boolean> -NewVariableName <String> -  
NewVariableValue <String> -VariableName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetByIdMandatory

```
Set-CMDeviceVariable -ResourceId <String> -VariableName <String> [-IsMask <Boolean> ] [-  
NewVariableName <String> ] [-NewVariableValue <String> ] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SetByValueMandatory

```
Set-CMDeviceVariable -Device <IResultObject> -VariableName <String> [-IsMask <Boolean> ] [-  
NewVariableName <String> ] [-NewVariableValue <String> ] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Detailed Description

The **Set-CMDeviceVariable** cmdlet modifies a device variable. Individual devices have device variables. Task sequence processing uses device variables.

Parameters

-Device<IResultObject>

Specifies a **CMDevice** object. To obtain a **CMDevice** object, use the **Get-CMDevice** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-DeviceName<String>

Specifies a device name. You can specify a NetBIOS name or a fully qualified domain name (FQDN).

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-IsMask<Boolean>

Indicates whether a value displays in the Configuration Manager console.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-NewVariableName<String>

Specifies a new name for the variable.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-NewVariableValue<String>

Specifies a new value for the variable.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ResourceId<String>

Specifies a Systems Management Server (SMS) ID.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-VariableName<String>

Specifies the name of the device variable.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

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

[Get-CMDeviceVariable](#)

[New-CMDeviceVariable](#)

[Remove-CMDeviceVariable](#)

Set-CMDiscoveryMethod

Set-CMDiscoveryMethod

Changes configuration settings of a discovery method.

Syntax

Parameter Set: SearchByActiveDirectoryForestDiscovery

```
Set-CMDiscoveryMethod -ActiveDirectoryForestDiscovery -SiteCode <String> [-  
EnableActiveDirectorySiteBoundaryCreation <Boolean> ] [-Enabled <Boolean> ] [-  
EnableSubnetBoundaryCreation <Boolean> ] [-PollingSchedule <IResultObject> ] [-Confirm] [-  
WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByActiveDirectoryGroupDiscovery

```
Set-CMDiscoveryMethod -ActiveDirectoryGroupDiscovery -SiteCode <String> [-  
DeltaDiscoveryIntervalMinutes <Int32> ] [-DiscoverDistributionGroupsMembership <Boolean> ]  
[-Enabled <Boolean> ] [-EnableDeltaDiscovery <Boolean> ] [-EnableFilteringExpiredLogon  
<Boolean> ] [-EnableFilteringExpiredPassword <Boolean> ] [-PollingSchedule <IResultObject> ]  
[-TimeSinceLastLogonDays <Int32> ] [-TimeSinceLastPasswordUpdateDays <Int32> ] [-Confirm] [-  
WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByActiveDirectorySystemDiscovery

```
Set-CMDiscoveryMethod -ActiveDirectorySystemDiscovery -SiteCode <String> [-  
ActiveDirectoryContainer <String[]> ] [-AddAdditionalAttribute <String[]> ] [-  
DeltaDiscoveryIntervalMinutes <Int32> ] [-Enabled <Boolean> ] [-EnableDeltaDiscovery  
<Boolean> ] [-EnableFilteringExpiredLogon <Boolean> ] [-EnableFilteringExpiredPassword  
<Boolean> ] [-PollingSchedule <IResultObject> ] [-RemoveAdditionalAttribute <String[]> ] [-  
TimeSinceLastLogonDays <Int32> ] [-TimeSinceLastPasswordUpdateDays <Int32> ] [-Confirm] [-  
WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByActiveDirectoryUserDiscovery

```
Set-CMDiscoveryMethod -ActiveDirectoryUserDiscovery -SiteCode <String> [-  
ActiveDirectoryContainer <String[]> ] [-AddAdditionalAttribute <String[]> ] [-  
DeltaDiscoveryIntervalMinutes <Int32> ] [-Enabled <Boolean> ] [-EnableDeltaDiscovery  
<Boolean> ] [-PollingSchedule <IResultObject> ] [-RemoveAdditionalAttribute <String[]> ] [-  
Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByHeartbeat

```
Set-CMDiscoveryMethod -Heartbeat -SiteCode <String> [-Enabled <Boolean> ] [-PollingSchedule  
<IResultObject> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByNetworkDiscovery

```
Set-CMDiscoveryMethod -NetworkDiscovery -SiteCode <String> [-Enabled <Boolean> ] [-  
NetworkDiscoveryType {ToplogyAndClient | ToplogyClientAndClientOperatingSystem | Topology} ]  
[-SlowNetworkSpeed <Boolean> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Set-CMDiscoveryMethod** cmdlet changes configuration settings of a discovery method. Discovery identifies computer and user resources that Microsoft System Center 2012 Configuration Manager can manage. When System Center 2012 Configuration Manager discovers a resource, System Center 2012 Configuration Manager creates a record in the Configuration Manager database for the resource and its associated information. You can then use the discovery information to help you to install the System Center 2012 Configuration Manager client and create custom queries and collections to logically group resources for related management tasks.

Parameters

-ActiveDirectoryContainer<String[]>

Specifies an array of names of Active Directory containers.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

-ActiveDirectoryForestDiscovery

Indicates that the discovery method discovers security groups, including local, global, and universal groups from specified locations in Active Directory Domain Services (AD DS).

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ActiveDirectoryGroupDiscovery

Indicates that the discovery method discovers additional information, including the computer organizational unit (OU) and group membership, about previously discovered computers from specified locations in AD DS.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ActiveDirectorySystemDiscovery

Indicates that the discovery method discovers computers from specified locations in AD DS.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ActiveDirectoryUserDiscovery

Indicates that the discovery method discovers users from specified locations in AD DS.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AddAdditionalAttribute<String[]>

Specifies an array of Active Directory object attributes. The cmdlet adds these attributes to the list of attributes that Configuration Manager discovers.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeltaDiscoveryIntervalMinutes<Int32>

Specifies how often, in minutes, to discover resources created or modified in AD DS since the last discovery cycle. To enable this feature, specify a value of \$True for the *EnableDeltaDiscovery* parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DiscoverDistributionGroupsMembership<Boolean>

Indicates whether Configuration Manager discovers the membership of distribution groups.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnableActiveDirectorySiteBoundaryCreation<Boolean>

Indicates whether Configuration Manager creates Active Directory boundaries from AD DS discovery information.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Enabled<Boolean>

Indicates whether to enable the Configuration Manager discovery. If you specify a value of \$False, Configuration Manager does not discover resources by using this discovery.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnableDeltaDiscovery<Boolean>

Indicates whether Configuration Manager discovers resources created or modified in AD DS since the last discovery cycle. If you specify a value of \$True for this parameter, specify a value for the *DeltaDiscoveryIntervalMinutes* parameter.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnableFilteringExpiredLogon<Boolean>

Indicates whether Configuration Manager discovers only computers that have logged onto a domain within a specified number of days. Specify the number of days by using the *TimeSinceLastLogonDays* parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnableFilteringExpiredPassword<Boolean>

Indicates whether Configuration Manager discovers only computers that have updated their computer account password within a specified number of days. Specify the number of days by using the *TimeSinceLastPasswordUpdateDays* parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnableSubnetBoundaryCreation<Boolean>

Indicates whether Configuration Manager creates IP address range boundaries from AD DS discovery information.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Heartbeat

Indicates that the discovery method updates discovery records for Configuration Manager clients in the Configuration Manager database without discovering new resources.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-NetworkDiscovery

Indicates that the discovery method searches the network infrastructure for network devices, such as printers, routers, and bridges, that have IP addresses.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-NetworkDiscoveryType<NetworkDiscoveryType>

Specifies the network discovery type. If you specify the *NetworkDiscovery* parameter, specify one of the following types:

- ToplogyAndClient. The discovery finds the topology of your network and potential client devices.
- ToplogyClientAndClientOperatingSystem. The discovery finds the topology of your network. The discovery finds potential client devices and their operating systems and versions.
- Topology. The discovery finds the topology of your network by discovering IP subnets and routers.

The acceptable values for this parameter are:

ToplogyAndClient	
ToplogyClientAndClientOperatingSystem	
Topology	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PollingSchedule<IResultObject>

Specifies a schedule object. To obtain a schedule object, use the **New-CMSchedule** cmdlet. The polling schedule determines how often Configuration Manager attempts to discover groups, systems, or user data.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RemoveAdditionalAttribute<String[]>

Specifies an array of Active Directory object attributes. The cmdlet removes these attributes from the list of attributes that Configuration Manager discovers.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteCode<String>

Specifies the site code for a Configuration Manager site.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SlowNetworkSpeed<Boolean>

Indicates whether Configuration Manager makes adjustments to its discovery settings for networks that have low bandwidth.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-TimeSinceLastLogonDays<Int32>

Specifies the number of days since the last logon when the *EnableFilteringExpiredLogon* parameter had a value of \$True.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-TimeSinceLastPasswordUpdateDays<Int32>

Specifies the number of days since that last password updated when the *EnableFilteringExpiredPassword* parameter had a value of \$True.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Modify network discovery

This command modifies network discovery for the site that has the site code CM4. The command specifies topology and client network discovery and the slow network speed option. The command also enables discovery.

```
PS C:\> Set-CMDiscoveryMethod -NetworkDiscovery -SiteCode "CM4" -Enabled $True -  
NetworkDiscoveryType ToplogyAndClient -SlowNetworkSpeed $True
```

Example 2: Modify Active Directory system discovery

The first command creates a schedule object by using the **New-CMSchedule** cmdlet and stores it in the `$Schedule` variable.

The second command enables the computer discovery for the site that has the site code CM4. The command specifies the schedule object stored in the `$Schedule` variable as the polling schedule and enables delta discovery to find new and modified computers since the last discovery. The command specifies that delta discovery takes place every 8 minutes.

The second command also limits the computers found to those that a user has logged onto in the last 80 days. Also, the command adds and removes specified attributes from the attributes used to limit computers.

```
PS C:\> $Schedule = New-CMSchedule -RecurInterval Minutes -Start "2012/10/20 00:00:00" -End "2013/10/20 00:00:00" -RecurCount 10
PS C:\> Set-CMDiscoveryMethod -ActiveDirectorySystemDiscovery -SiteCode "CM4" -
AddAdditionalAttribute "331", "431", "134" -DeltaDiscoveryIntervalMinutes 8 -Enabled $True -
EnableDeltaDiscovery $True -EnableFilteringExpiredLogon $True -PollingSchedule $Schedule -
RemoveAdditionalAttribute "123","cn" -TimeSinceLastLogonDays 80
```

Example 3: Modify forest discovery

The first command creates a schedule object by using the **New-CMSchedule** cmdlet, and then stores it in the `$Schedule` variable.

The second command enables this discovery site that has the site code CM4. The command specifies the schedule object stored in the `$Schedule` variable as the polling interval and enables Active Directory boundary creation and subnet boundary creation.

```
PS C:\> $Schedule = New-CMSchedule -RecurInterval Minutes -Start "2012/10/20 00:00:00" -End "2013/10/20 00:00:00" -RecurCount 10
PS C:\> Set-CMDiscoveryMethod -ActiveDirectoryForestDiscovery -SiteCode "CM4" -
EnableActiveDirectorySiteBoundaryCreation $True -Enabled $True -
EnableSubnetBoundaryCreation $True -PollingSchedule $Schedule
```

Example 4: Enable heartbeat discovery

The first command creates a schedule object by using the **New-CMSchedule** cmdlet and stores it in the `$Schedule` variable.

The second command enables heartbeat discovery and specifies the object stored in the `$Schedule` variable as the polling schedule for the site that has the site code CM4.

```
PS C:\> $Schedule = New-CMSchedule -RecurInterval Minutes -Start "2012/10/20 00:00:00" -End "2013/10/20 00:00:00" -RecurCount 10
PS C:\> Set-CMDiscoveryMethod -Heartbeat -SiteCode "CM4" -Enabled $True -PollingSchedule $Schedule
```

Related topics

[Get-CMDiscoveryMethod](#)

[New-CMSchedule](#)

Set-CMDistributionPoint

Set-CMDistributionPoint

Modifies a Configuration Manager distribution point on a site system server.

Syntax

Parameter Set: SetByName

```
Set-CMDistributionPoint -SiteCode <String> -SiteSystemServerName <String> [-AddBoundaryGroupName <String[]> ] [-AllowFallbackForContent <Boolean> ] [-AllowPreStaging <Boolean> ] [-AllowRespondIncomingPxeRequest <Boolean> ] [-CertificateExpirationTimeUtc <DateTime> ] [-CertificatePassword <SecureString> ] [-CertificatePath <String> ] [-ClientCommunicationType {HTTP | HTTPS} ] [-ClientConnectionType {Internet | InternetAndIntranet | Intranet} ] [-ClientTransferRate {None | Profile100Mbps | Profile10Mbps | Profile1Gbps | ProfileCustom} ] [-ComputersUsePxePassword <SecureString> ] [-ContentMonitoringPriority {High | Highest | Low | Lowest | Medium} ] [-EnableAnonymous <Boolean> ] [-EnableMulticast <Boolean> ] [-EnablePullDP <Boolean> ] [-EnablePxeSupport <Boolean> ] [-EnableScheduledMultiCast <Boolean> ] [-EnableUnknownComputerSupport <Boolean> ] [-EnableValidateContent <Boolean> ] [-EndIpAddress <String> ] [-EndUdpPort <Int32> ] [-MacAddressForRespondingPxeRequest <String[]> ] [-MinimumSessionSize <Int32> ] [-MulticastMaximumClientCount <Int32> ] [-PxeServerResponseDelaySeconds <Int32> ] [-RemoveBoundaryGroupName <String[]> ] [-RespondToAllNetwork] [-SessionStartDelayMinutes <Int32> ] [-SourceDistributionPoints <String[]> ] [-SourceDPRanks <Int32[]> ] [-StartIpAddress <String> ] [-StartUdpPort <Int32> ] [-UseAnyRangeIP] [-UseComputerAccount] [-UserDeviceAffinity {AllowWithAutomaticApproval | AllowWithManualApproval | DoNotUse} ] [-UserName <String> ] [-ValidateContentSchedule <IResultObject> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetByValue

```
Set-CMDistributionPoint -InputObject <IResultObject> [-AddBoundaryGroupName <String[]> ] [-AllowFallbackForContent <Boolean> ] [-AllowPreStaging <Boolean> ] [-AllowRespondIncomingPxeRequest <Boolean> ] [-CertificateExpirationTimeUtc <DateTime> ] [-CertificatePassword <SecureString> ] [-CertificatePath <String> ] [-ClientCommunicationType {HTTP | HTTPS} ] [-ClientConnectionType {Internet | InternetAndIntranet | Intranet} ] [-ClientTransferRate {None | Profile100Mbps | Profile10Mbps | Profile1Gbps | ProfileCustom} ] [-ComputersUsePxePassword <SecureString> ] [-ContentMonitoringPriority {High | Highest | Low | Lowest | Medium} ] [-EnableAnonymous <Boolean> ] [-EnableMulticast <Boolean> ] [-EnablePullDP <Boolean> ] [-EnablePxeSupport <Boolean> ] [-EnableScheduledMultiCast <Boolean> ] [-EnableUnknownComputerSupport <Boolean> ] [-EnableValidateContent <Boolean> ] [-EndIpAddress <String> ] [-EndUdpPort <Int32> ] [-MacAddressForRespondingPxeRequest <String[]> ] [-MinimumSessionSize <Int32> ] [-MulticastMaximumClientCount <Int32> ] [-PxeServerResponseDelaySeconds <Int32> ] [-RemoveBoundaryGroupName <String[]> ] [-RespondToAllNetwork] [-SessionStartDelayMinutes <Int32> ] [-SourceDistributionPoints <String[]> ] [-SourceDPRanks <Int32[]> ] [-StartIpAddress <String> ] [-StartUdpPort <Int32> ] [-UseAnyRangeIP] [-UseComputerAccount] [-UserDeviceAffinity {AllowWithAutomaticApproval | AllowWithManualApproval | DoNotUse} ] [-UserName <String> ] [-ValidateContentSchedule <IResultObject> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetSecurityScopeById
Set-CMDistributionPoint -Id <String> -SecurityScopeAction {AddMembership | RemoveMembership}
-SecurityScopeName <String> [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SetSecurityScopeByName
Set-CMDistributionPoint -Name <String> -SecurityScopeAction {AddMembership |
RemoveMembership} -SecurityScopeName <String> [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SetSecurityScopeByValue
Set-CMDistributionPoint -InputObject <IResultObject> -SecurityScopeAction {AddMembership |
RemoveMembership} -SecurityScopeName <String> [-Confirm] [-WhatIf] [<CommonParameters>]

Detailed Description

The **Set-CMDistributionPoint** cmdlet modifies a distribution point on a site system server. A distribution point is a site system role that Microsoft System Center 2012 Configuration Manager uses to store files for clients to download, such as application content, software packages, software updates, operating system images, and boot images.

Parameters

-AddBoundaryGroupName<String[]>

Specifies an array of names of boundary groups to add to the distribution point.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AllowFallbackForContent<Boolean>

Indicates whether clients can use a fallback source location for content.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AllowPreStaging<Boolean>

Indicates whether the distribution point can pre-stage contents.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AllowRespondIncomingPxeRequest<Boolean>

Indicates whether the distribution point can respond to pre-boot execution environment (PXE) requests.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CertificateExpirationTimeUtc<DateTime>

Specifies the date and time when the certificate expires as Coordinated Universal Time (UTC).

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CertificatePassword<SecureString>

Specifies the password, as a secure string, for the public key infrastructure (PKI) client certificate for the distribution point.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CertificatePath<String>

Specifies the import path for the PKI issued certificate that the distribution point uses.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ClientCommunicationType<ComputerCommunicationType>

Specifies the client communication type. Valid values are HTTP and HTTPS.

The acceptable values for this parameter are:

HTTP	
HTTPS	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ClientConnectionType<ClientConnectionTypes>

Specifies the type of connection that clients use. Valid values are:

- Internet
- InternetAndIntranet
- Intranet

The acceptable values for this parameter are:

Internet	
InternetAndIntranet	
Intranet	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ClientTransferRate<NetworkProfile>

Specifies the client transfer rate. Valid values are:

- None
- Profile100Mbps
- Profile10Mbps
- Profile1Gbps
- ProfileCustom

The acceptable values for this parameter are:

None	
Profile100Mbps	
Profile10Mbps	
Profile1Gbps	
ProfileCustom	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ComputersUsePxePassword<SecureString>

Specifies the password, as a secure string, for computers that use PXE.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ContentMonitoringPriority<Priority>

Specifies the content monitoring priority. Valid values are:

- High
- Highest
- Low
- Lowest
- Medium

The acceptable values for this parameter are:

High	
Highest	
Low	
Lowest	
Medium	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnableAnonymous<Boolean>

Indicates whether the distribution point permits anonymous connections from Configuration Manager clients to the content library.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnableMulticast<Boolean>

Indicates whether multicast is enabled on the distribution point.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnablePullDP<Boolean>

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnablePxeSupport<Boolean>

Indicates whether PXE is enabled on the distribution point.

When you enable PXE, Configuration Manager installs Windows Deployment Services on the server, if required. Windows Deployment Service is the service that performs the PXE boot to install operating systems. After you create the distribution point, Configuration Manager installs a provider in Windows Deployment Services that uses the PXE boot functions.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnableScheduledMultiCast<Boolean>

Indicates whether you can configure when Configuration Manager deploys the operating system image from the distribution point.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnableUnknownComputerSupport<Boolean>

Indicates whether support for unknown computers is enabled. Unknown computers are computers that are not managed by Configuration Manager.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnableValidateContent<Boolean>

Indicates whether the distribution point validates the integrity of the content files in the package.

Aliases	none
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EndIpAddress<String>

Specifies the ending IP address in the range of IP addresses that Configuration Manager uses to send data to the destination computers.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EndUdpPort<Int32>

Specifies the ending port in the range of user datagram protocol (UDP) ports that Configuration Manager uses to send data to the destination computers.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Id<String>

Specifies the ID of a distribution point.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a distribution point object. To obtain a distribution point object, use the **Get-CMDistributionPoint** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-MacAddressForRespondingPxeRequest<String[]>

Specifies an array of media access controller (MAC) addresses that the distribution point uses to respond to pre-boot execution environment (PXE) requests.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-MinimumSessionSize<Int32>

Specifies how many requests must be received before Configuration Manager starts to deploy the operating system.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-MulticastMaximumClientCount<Int32>

Specifies the maximum number of destination computers that can download the operating system from this distribution point.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies the name of a computer that hosts a distribution point.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PxeServerResponseDelaySeconds<Int32>

Specifies how long, in seconds, the distribution point delays before it responds to computer requests when you are using multiple PXE-enabled distribution points. By default, the Configuration Manager PXE service point responds first to network PXE requests.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RemoveBoundaryGroupName<String[]>

Specifies an array of names of boundary groups to remove from the distribution point.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RespondToAllNetwork

Indicates that the distribution point responds to PXE requests that arrive on any of its network interfaces.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SecurityScopeAction<SecurityScopeActionType>

Specifies an action for a security scope. Valid values are: AddMembership and RemoveMembership.

The acceptable values for this parameter are:

AddMembership	
RemoveMembership	

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SecurityScopeName<String>

Specifies the name of a security scope. A security scope name can be Default or the name of a custom security scope.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SessionStartDelayMinutes<Int32>

Specifies the number of minutes that Configuration Manager waits before it responds to the first deployment request.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteCode<String>

Specifies the site code for the Configuration Manager site that hosts this site system role.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteSystemServerName<String>

Specifies the name of a server that hosts a site system role.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SourceDistributionPoints<String[]>

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SourceDPRanks<Int32[]>

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-StartIpAddress<String>

Specifies the starting IP address in the range of IP addresses that Configuration Manager uses to send data to the destination computers.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-StartUdpPort<Int32>

Specifies the starting port in the range of UDP ports that Configuration Manager uses to send data to the destination computers.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UseAnyRangeIP

Indicates that multicast uses addresses within any range.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UseComputerAccount

Indicates that the distribution point uses its account as the multicast connection account when it connects to the primary site database.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UserDeviceAffinity<UserDeviceAffinityType>

Specifies how you want the distribution point to associate users with the destination computer for PXE deployments. Valid values are:

- AllowWithAutomaticApproval
- AllowWithManualApproval
- DoNotUse

The acceptable values for this parameter are:

AllowWithAutomaticApproval	
AllowWithManualApproval	
DoNotUse	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UserName<String>

Specifies the name of the user that the distribution site system components use to connect to the primary site database.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ValidateContentSchedule<IResultObject>

Specifies a **CMSchedule** object. A **CMSchedule** object defines the schedule for validating the integrity of content files on the distribution point. To create a **CMSchedule** object, use the **New-CMSchedule** cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Add a security scope to a distribution point by using role name

This command adds the security scope named Scope22 to a distribution point. The computer named CMDistribution.Western.Contoso.com hosts the distribution point.

```
PS C:\> Set-CMDistributionPoint -Name "CMDistribution.Western.Contoso.com" -  
SecurityScopeAction AddMembership -SecurityScopeName "Scope22"
```

Example 2: Remove a security scope by using role ID

This command removes a security scope named Scope22 from the distribution point that has the specified ID.

```
PS C:\> Set-CMDistributionPoint -Id "SMS00073" -SecurityScopeAction RemoveMembership -  
SecurityScopeName "Scope22"
```

Related topics

[Add-CMDistributionPoint](#)

[Get-CMDistributionPoint](#)

[Remove-CMDistributionPoint](#)

[Update-CMDistributionPoint](#)

[New-CMSchedule](#)

Set-CMDistributionPointGroup

Set-CMDistributionPointGroup

Changes the configuration settings of distribution point groups.

Syntax

Parameter Set: SetById

```
Set-CMDistributionPointGroup -Id <String[]> [-Description <String> ] [-NewName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetByName

```
Set-CMDistributionPointGroup -Name <String> [-Description <String> ] [-NewName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetByValueMandatory

```
Set-CMDistributionPointGroup -InputObject <IResultObject> [-Description <String> ] [-NewName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetSecurityScopeById

```
Set-CMDistributionPointGroup -Id <String[]> -SecurityScopeAction {AddMembership | RemoveMembership} -SecurityScopeName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetSecurityScopeByName

```
Set-CMDistributionPointGroup -Name <String> -SecurityScopeAction {AddMembership | RemoveMembership} -SecurityScopeName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetSecurityScopeByValue

```
Set-CMDistributionPointGroup -InputObject <IResultObject> -SecurityScopeAction {AddMembership | RemoveMembership} -SecurityScopeName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Set-CMDistributionPointGroup** cmdlet changes the configuration settings of one or more distribution point groups.

Parameters

-Description<String>

Specifies a description for the distribution point group.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Id<String[]>

Specifies an array of IDs for distribution point groups.

Aliases	GroupId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a **CMDistributionPointGroup** object. To obtain a **CMDistributionPointGroup** object, use the [Get-CMDistributionPointGroup](#) cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies a name of a distribution point group.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-NewName<String>

Specifies a new name for the distribution point group.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SecurityScopeAction<SecurityScopeActionType>

Specifies an action for a security scope. Valid values are: AddMembership and RemoveMembership.

The acceptable values for this parameter are:

AddMembership	
RemoveMembership	

Aliases	none
Required?	true
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SecurityScopeName<String>

Specifies the name of a security scope. A security scope name can be Default or the name of a custom security scope.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Add a distribution point group to a security scope

This command adds the distribution point group as a member of the security scope named CScope02.

```
PS C:\> Set-CMDistributionPointGroup -SecurityScopeAction AddMembership -SecurityScopeName "CScope02" -Name "DpgDept01"
```

Example 2: Remove a distribution point group from a security scope

This command removes the distribution point group named DpgDept01 as a member of the security scope named CScope02.

```
PS C:\> Set-CMDistributionPointGroup -SecurityScopeAction RemoveMembership -SecurityScopeName "CScope02" -Name "DpgDept01"
```

Related topics

[Get-CMDistributionPointGroup](#)

[New-CMDistributionPointGroup](#)

[Remove-CMDistributionPointGroup](#)

Set-CMDriver

Set-CMDriver

Changes the settings of a device driver.

Syntax

Parameter Set: SetDriverById

```
Set-CMDriver -Id <String> [-AddBootImagePackage <IResultObject[]> ] [-AddDriverPackage <IResultObject[]> ] [-AdministrativeCategory <IResultObject[]> ] [-Description <String> ] [-DriverSource <String> ] [-EnableAndAllowInstall <Boolean> ] [-NewName <String> ] [-RemoveBootImagePackage <IResultObject[]> ] [-RemoveDriverPackage <IResultObject[]> ] [-RunOnAnyPlatform] [-SupportedPlatformName <String[]> ] [-UpdateDistributionPointsforBootImagePackage <Boolean> ] [-UpdateDistributionPointsforDriverPackage <Boolean> ] [-Confirm] [-WhatIf] [<CommonParameters>]
```

Parameter Set: SetDriverByName

```
Set-CMDriver -Name <String> [-AddBootImagePackage <IResultObject[]> ] [-AddDriverPackage <IResultObject[]> ] [-AdministrativeCategory <IResultObject[]> ] [-Description <String> ] [-DriverSource <String> ] [-EnableAndAllowInstall <Boolean> ] [-NewName <String> ] [-RemoveBootImagePackage <IResultObject[]> ] [-RemoveDriverPackage <IResultObject[]> ] [-RunOnAnyPlatform] [-SupportedPlatformName <String[]> ] [-UpdateDistributionPointsforBootImagePackage <Boolean> ] [-UpdateDistributionPointsforDriverPackage <Boolean> ] [-Confirm] [-WhatIf] [<CommonParameters>]
```

Parameter Set: SetDriverByResultObject

```
Set-CMDriver -InputObject <IResultObject> [-AddBootImagePackage <IResultObject[]> ] [-AddDriverPackage <IResultObject[]> ] [-AdministrativeCategory <IResultObject[]> ] [-Description <String> ] [-DriverSource <String> ] [-EnableAndAllowInstall <Boolean> ] [-NewName <String> ] [-RemoveBootImagePackage <IResultObject[]> ] [-RemoveDriverPackage <IResultObject[]> ] [-RunOnAnyPlatform] [-SupportedPlatformName <String[]> ] [-UpdateDistributionPointsforBootImagePackage <Boolean> ] [-UpdateDistributionPointsforDriverPackage <Boolean> ] [-Confirm] [-WhatIf] [<CommonParameters>]
```

Detailed Description

The **Set-CMDriver** cmdlet changes settings of a device driver in the driver catalog.

Parameters

-AddBootImagePackage<IResultObject[]>

Specifies an array of boot image objects. Use this parameter to specify the boot images that can install the device drivers. To obtain a boot image object, use the **Get-CMBootImage** cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AddDriverPackage<IResultObject[]>

Specifies an array of driver package objects. Use this parameter to specify the driver packages that Configuration Manager uses to distribute the device drivers. To obtain a driver package object, use the **Get-CMDriverPackage** cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AdministrativeCategory<IResultObject[]>

Specifies an array of administrative categories. Assign the device drivers to an administrative category for filtering purposes, such as Desktops or Notebooks categories.

To obtain an administrative category object, use the **Get-CMCategory** cmdlet.

Aliases	none
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Description<String>

Specifies a description for the device driver.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DriverSource<String>

Specifies the driver package source location. When you create a driver package, the source location of the package must point to an empty network share that is not used by another driver package.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnableAndAllowInstall<Boolean>

Indicates whether Configuration Manager enables the drivers and allows computers to install the drivers.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Id<String>

Specifies the ID of a device driver.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a driver object. To obtain a driver object, use the **Get-CMDriver** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue, ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies the name of a device driver.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-NewName<String>

Specifies a new name for the device driver.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RemoveBootImagePackage<IResultObject[]>

Specifies an array of boot image objects. Use this parameter to remove the boot images that can install the device driver. To obtain a boot image object, use the **Get-CMBootImage** cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RemoveDriverPackage<IResultObject[]>

Specifies an array of driver package objects. Use this parameter to remove the driver packages that Configuration Manager uses to distribute the device drivers. To obtain a driver package object, use the **Get-CMDriverPackage** cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RunOnAnyPlatform

Indicates that the device driver can run on all platforms.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SupportedPlatformName<String[]>

Specifies an array of names of platforms on which the device driver can run.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UpdateDistributionPointsforBootImagePackage<Boolean>

Indicates whether Configuration Manager updates distribution points when the device driver is added to the boot image package.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UpdateDistributionPointsforDriverPackage<Boolean>

Indicates whether Configuration Manager updates distribution points when the device driver is added to the driver package.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Changes the settings of a device driver

The first command uses the **Get-CMBootImage** cmdlet to get the boot image that has the ID CM100004, and stores the result in the \$BootPackage variable.

The second command changes the settings of the device driver named Fabrikam Digital Audio. The command specifies a new name and a description for the device driver. The command enables the driver and allows computers to install it. The command specifies that the device driver can run on all platforms, and removes the boot image stored in \$BootPackage that can install the device driver.

```
PS C:\> $BootPackage = Get-CMBootImage -Id "CM100004"
PS C:\> Set-CMDriver -Name "Fabrikam Digital Audio" -NewName "FabrikamDA" -Description
"Previous version" -EnableAndAllowInstall $True -RunOnAnyPlatform $True -
RemoveBootImagePackage $BootPackage
```

Related topics

[Get-CMDriver](#)

[Import-CMDriver](#)

[Enable-CMDriver](#)

[Disable-CMDriver](#)

[Remove-CMDriver](#)

[Get-CMCategory](#)

[Get-CMBootImage](#)

[Get-CMDriverPackage](#)

Set-CMDriverBootImage

Set-CMDriverBootImage

Adds a driver to a boot image or removes a driver from a boot image.

Syntax

Parameter Set: SetDriverBootImagesById_Id

```
Set-CMDriverBootImage -BootImageId <String> -DriverId <String> -SetDriveBootImageAction  
{AddDriverToBootImage | RemoveDriverFromBootImage} [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SetDriverBootImagesById_Name

```
Set-CMDriverBootImage -BootImageName <String> -DriverId <String> -SetDriveBootImageAction  
{AddDriverToBootImage | RemoveDriverFromBootImage} [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SetDriverBootImagesById_Object

```
Set-CMDriverBootImage -BootImage <IResultObject> -DriverId <String> -SetDriveBootImageAction  
{AddDriverToBootImage | RemoveDriverFromBootImage} [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SetDriverBootImagesByName_Id

```
Set-CMDriverBootImage -BootImageId <String> -DriverName <String> -SetDriveBootImageAction  
{AddDriverToBootImage | RemoveDriverFromBootImage} [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SetDriverBootImagesByName_Name

```
Set-CMDriverBootImage -BootImageName <String> -DriverName <String> -SetDriveBootImageAction  
{AddDriverToBootImage | RemoveDriverFromBootImage} [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SetDriverBootImagesByName_Object

```
Set-CMDriverBootImage -BootImage <IResultObject> -DriverName <String> -  
SetDriveBootImageAction {AddDriverToBootImage | RemoveDriverFromBootImage} [-Confirm] [-  
WhatIf] [ <CommonParameters>]
```

Parameter Set: SetDriverBootImagesByObject_Id

```
Set-CMDriverBootImage -BootImageId <String> -Driver <IResultObject> -SetDriveBootImageAction  
{AddDriverToBootImage | RemoveDriverFromBootImage} [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SetDriverBootImagesByObject_Name

```
Set-CMDriverBootImage -BootImageName <String> -Driver <IResultObject> -  
SetDriveBootImageAction {AddDriverToBootImage | RemoveDriverFromBootImage} [-Confirm] [-  
WhatIf] [ <CommonParameters>]
```

```
Parameter Set: SetDriverBootImagesByObject_Object
Set-CMDriverBootImage -BootImage <IResultObject> -Driver <IResultObject> -
SetDriveBootImageAction {AddDriverToBootImage | RemoveDriverFromBootImage} [-Confirm] [-
WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Set-CMDriverBootImage** cmdlet adds a driver to a boot image or removes a driver from a boot image. You can add Windows device drivers that you have imported into the Microsoft System Center 2012 Configuration Manager driver catalog to one or more boot images. You should add only mass storage device drivers and network adapter device drivers to boot images because other types of drivers are not needed and will increase the size of the boot image.

Parameters

-BootImage<IResultObject>

Specifies a **CMBootImage** object. To obtain a **CMBootImage** object, use the [Get-CMBootImage](#) cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-BootImageId<String>

Specifies the ID of a boot image.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-BootImageName<String>

Specifies the name of a boot image.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Driver<IResultObject>

Specifies a driver object. To obtain a **CMDriver** object, use the [Get-CMDriver](#) cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DriverId<String>

Specifies the ID of a driver.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-DriverName<String>

Specifies the name of a driver.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SetDriveBootImageAction<SetDriveBootImageActionType>

Specifies the boot image action. Valid values are:

- AddDriverToBootImage
- RemoveDriverFromBootImage

The acceptable values for this parameter are:

AddDriverToBootImage	
RemoveDriverFromBootImage	

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Add a driver to a boot image

This command adds the driver named Adaptec Embedded SCSI HostRAID Controller to the boot image named Boot image (x64).

```
PS C:\> Set-CMDriverBootImage -SetDriveBootImageAction AddDriverToBootImage -DriverName  
"Adaptec Embedded SCSI HostRAID Controller" -BootImageName "Boot image (x64)"
```

Example 2: Remove a driver from a boot image

This command removes the driver named Adaptec SCSI HostRAID Management Processor Device from the boot image named Boot image (x64).

```
PS C:\> Set-CMDriverBootImage -SetDriveBootImageAction RemoveDriverFromBootImage -DriverName "Adaptec SCSI HostRAID Management Processor Device" -BootImageName "Boot image (x64)"
```

Related topics

[Get-CMBootImage](#)

[Get-CMDriver](#)

Set-CMDriverPackage

Set-CMDriverPackage

Modifies a driver package

Syntax

Parameter Set: SetById

```
Set-CMDriverPackage -Id <String[]> [-Description <String> ] [-DriverPackageSource <String> ]  
[-NewName <String> ] [-SecuredScopeNames <String> ] [-Version <String> ] [-Confirm] [-  
WhatIf] [ <CommonParameters>]
```

Parameter Set: SetByName

```
Set-CMDriverPackage -Name <String> [-Description <String> ] [-DriverPackageSource <String> ]  
[-NewName <String> ] [-SecuredScopeNames <String> ] [-Version <String> ] [-Confirm] [-  
WhatIf] [ <CommonParameters>]
```

Parameter Set: SetByValueMandatory

```
Set-CMDriverPackage -InputObject <IResultObject> [-Description <String> ] [-  
DriverPackageSource <String> ] [-NewName <String> ] [-Version <String> ] [-Confirm] [-  
WhatIf] [ <CommonParameters>]
```

Parameter Set: SetSecurityScopeById

```
Set-CMDriverPackage -Id <String[]> -SecurityScopeAction {AddMembership | RemoveMembership} -  
SecurityScopeName <String> [-SecuredScopeNames <String> ] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SetSecurityScopeByName

```
Set-CMDriverPackage -Name <String> -SecurityScopeAction {AddMembership | RemoveMembership} -  
SecurityScopeName <String> [-SecuredScopeNames <String> ] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SetSecurityScopeByValue

```
Set-CMDriverPackage -InputObject <IResultObject> -SecurityScopeAction {AddMembership |  
RemoveMembership} -SecurityScopeName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Set-CMDriverPackage** cmdlet modifies a driver package in Microsoft System Center 2012 Configuration Manager.

Parameters

-Description<String>

Specifies a description of a driver package.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DriverPackageSource<String>

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Id<String[]>

Specifies an array of identifiers for a driver package.

Aliases	PackageId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a driver package object. To obtain a driver package object, use the **Get-CMDriverPackage** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies a name of a driver package.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-NewName<String>

Specifies a new name of a driver package.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SecuredScopeNames<String>

Specifies the names of security scopes. A security scope name can be Default or the name of a custom-created security scope.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SecurityScopeAction<SecurityScopeActionType>

Specifies an action for a security scope. Valid values are: AddMembership and RemoveMembership. The acceptable values for this parameter are:

AddMembership	
RemoveMembership	

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SecurityScopeName<String>

Specifies the name of a security scope. A security scope name can be Default or the name of a custom security scope.

Aliases	none
---------	------

Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Version<String>

Specifies a version of a security scope.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Add a security scope action to a driver package

This command adds a security scope action to the driver package that is named Windows 7 Standard Hardware Package.

```
PS C:\> Set-CMDriverPackage -SecurityScopeAction AddMembership -SecurityScopeName "Scope 001" -Name "Windows 7 Standard Hardware Package"
```

Example 2: Remove a security scope action from a driver package

This command removes a security scope action from the driver package that is named Windows 7 Standard Hardware Package.

```
PS C:\> Set-CMDriverPackage -SecurityScopeAction RemoveMembership -SecurityScopeName "Scope 001" -Name "Windows 7 Standard Hardware Package"
```

Related topics

[Export-CMDriverPackage](#)

[Get-CMDriverPackage](#)

[Import-CMDriverPackage](#)

[New-CMDriverPackage](#)

[Remove-CMDriverPackage](#)

Set-CMEmailNotificationComponent

Set-CMEmailNotificationComponent

Changes configuration settings of an email notification component.

Syntax

Parameter Set: Disable

```
Set-CMEmailNotificationComponent -DisableEmailNotification [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: Enable

```
Set-CMEmailNotificationComponent -EnableEmailNotification -SendFrom <String> -SmtpServerFqdn  
<String> -TypeOfAuthentication {Anonymous | DefaultServiceAccount | Other} [-Port <Int32> ]  
[-UserName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Set-CMEmailNotificationComponent** cmdlet changes configuration settings of an email notification component in Microsoft System Center 2012 Configuration Manager. You can configure the email notification component for each System Center 2012 Configuration Manager site to configure email subscriptions to alerts.

Parameters

-DisableEmailNotification

Indicates that email notification is disabled.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnableEmailNotification

Indicates that Configuration Manager uses an SMTP server to send email alerts.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Port<Int32>

Specifies the outgoing SMTP port for sending email alerts.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SendFrom<String>

Specifies the email address from which Configuration Manager sends email alerts.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SmtpServerFqdn<String>

Specifies the fully qualified domain name (FQDN) of the SMTP server.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-TypeOfAuthentication<AuthenticationMethod>

Specifies the method by which Configuration Manager authenticates the site server to the SMTP Server. Valid values are:

- Anonymous
- DefaultServiceAccount
- Other

The acceptable values for this parameter are:

Anonymous	
DefaultServiceAccount	
Other	

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UserName<String>

Specifies the user name to authenticate with the SMTP server from which Configuration Manager sends email alerts. This parameter also specifies the SMTP Server Connection account.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Enable email notification

This command enables email notification for System Center 2012 Configuration Manager. The command specifies that System Center 2012 Configuration Manager uses the System Center 2012 Configuration Manager site that has the site code CM2 on the server cmcen-dist02.tsqa.corp.contoso.com to host the site system role for email notification. The command specifies that System Center 2012 Configuration Manager uses the server named mail.corp.contosco.com for the email server and specifies the outgoing SMTP port for sending email alerts. The command specifies that System Center 2012 Configuration Manager uses the default service account for authenticating the site server to the SMTP Server, and specifies that System Center 2012 Configuration Manager sends email alerts from the email address evan.narvaez@contoso.com.

```
PS C:\> Set-CMEmailNotificationComponent -SiteSystemServerName "cmcen-  
dist02.tsqa.corp.contoso.com" -SiteCode "CM2" -EnableEmailNotification $True -MstpServerFqdn  
"mail.corp.contosco.com" -Port 25 -TypeOfAuthentication DefaultServiceAccount -SendFrom  
"evan.narvaez@contoso.com"
```

Example 2: Disable email notification

This command disables email notification for System Center 2012 Configuration Manager on the site server named cmcen-dist02.tsqa.corp.contoso.com.

```
PS C:\> Set-CMEmailNotificationComponent -SiteSystemServerName "cmcen-  
dist02.tsqa.corp.contoso.com" -EnableEmailNotification $False
```

Example 3: Set the outgoing SMTP port for email notification

This command sets the outgoing SMTP port that System Center 2012 Configuration Manager uses for sending email alerts on the site server named cmcen-dist02.tsqa.corp.contoso.com to port 27.

```
PS C:\> Set-CMEmailNotificationComponent -SiteSystemServerName "cmcen-  
dist02.tsqa.corp.contoso.com" -Port 27
```

Related topics

[Get-CMEmailNotificationComponent](#)

Set-CMEndpointProtectionPoint

Set-CMEndpointProtectionPoint

Modifies a site system role for Endpoint Protection.

Syntax

Parameter Set: SetByName

```
Set-CMEndpointProtectionPoint -ProtectionService {AdvancedMembership | BasicMembership | DoNotJoinMaps} -SiteCode <String> -SiteSystemServerName <String> [-Confirm] [-WhatIf] [<CommonParameters>]
```

Parameter Set: SetByValue

```
Set-CMEndpointProtectionPoint -InputObject <IResultObject> -ProtectionService {AdvancedMembership | BasicMembership | DoNotJoinMaps} [-Confirm] [-WhatIf] [<CommonParameters>]
```

Detailed Description

The **Set-CMEndpointProtectionPoint** cmdlet modifies a site system role for System Center 2012 Endpoint Protection in Microsoft System Center 2012 Configuration Manager.

Endpoint Protection lets you manage antimalware policies and Windows Firewall security for client computers in System Center 2012 Configuration Manager. In order to use Endpoint Protection with System Center 2012 Configuration Manager, you must install a single site system role for Endpoint Protection, either in the central site or in a stand-alone primary site. For more information about Endpoint Protection in System Center 2012 Configuration Manager, see the [Endpoint Protection in Configuration Manager](http://go.microsoft.com/fwlink/?LinkId=268427) (<http://go.microsoft.com/fwlink/?LinkId=268427>) on TechNet.

Parameters

-InputObject<IResultObject>

Specifies an input object. To obtain an input object, use the **Get-CMEndpointProtectionPoint** cmdlet.

Aliases	none
Required?	true
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ProtectionService<MapsMembershipType>

Specifies the type of membership to set for Microsoft Active Protection Service (MAPS). Valid values are:

- AdvancedMembership
- BasicMembership
- DoNotJoinMaps

The acceptable values for this parameter are:

AdvancedMembership	
BasicMembership	
DoNotJoinMaps	

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteCode<String>

Specifies the site code for a Configuration Manager site.

Aliases	none
Required?	true
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteSystemServerName<String>

Specifies the name of a server that hosts a site system role.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Set an endpoint protection point

The command sets the endpoint protection point for the server, and specifies the membership type for the *ProtectionService* parameter.

```
PS C:\> Set-CMEndpointProtectionPoint -SiteSystemServerName "CM-Contoso.Contoso.Com" -  
SiteCode "CM2" -ProtectionService AdvancedMembership
```

Example 2: Set an endpoint protection point by using an input object

The first command uses the **Get-CMEndpointProtectionPoint** cmdlet to get an endpoint protection point, and stores the result in the \$Epp variable.

The second command sets the endpoint protection point for the server by using the input object from the previous command.

```
PS C:\> $Epp = Get-CMEndpointProtectionPoint -SiteSystemServerName "CM-Contoso.Contoso.Com"  
-SiteCode "CM2"  
PS C:\> Set-CMEndpointProtectionPoint -InputObject $Epp -ProtectionService BasicMembership
```

Related topics

[Add-CMEndpointProtectionPoint](#)

[Get-CMEndpointProtectionPoint](#)

[Remove-CMEndpointProtectionPoint](#)

Set-CMEndpointProtectionSummarizationSchedule

Set-CMEndpointProtectionSummarizationSchedule

Modifies an Endpoint Protection summarization schedule.

Syntax

Parameter Set: Set

```
Set-CMEndpointProtectionSummarizationSchedule -Interval <Int32> [-Unit {Days | Hours | Minutes} ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Set-CMEndpointProtectionSummarizationSchedule** cmdlet modifies the settings of an System Center 2012 Endpoint Protection summarization schedule. For more information about Endpoint Protection summarization schedules, see [How to Monitor Endpoint Protection in Configuration Manager](http://go.microsoft.com/fwlink/?LinkId=268428) (<http://go.microsoft.com/fwlink/?LinkId=268428>) on TechNet.

Parameters

-Interval<Int32>

Specifies an amount of time, as an integer. This value works with the unit type you specify in the *Unit* parameter. Valid values for this parameter depend on the unit that you select:

- Minutes: 10 through 59.
- Hours: 1 through 23.
- Days: 1 through 31.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-Unit<SummarizationScheduleUnit>

Specifies a unit to use to define an interval for the summarization schedule. Valid values are:

- Days
- Hours
- Minutes

The acceptable values for this parameter are:

Days	
Hours	
Minutes	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Modify an Endpoint Protection summarization schedule

This command modifies the interval and unit values to specify that 10 days pass before the Endpoint Protection Summarization Schedule runs again.

```
PS C:\> Set-CMEndpointProtectionSummarizationSchedule -Interval 10 -UnitType "Days"
```

Related topics

[Get-CMEndpointProtectionSummarizationSchedule](#)

Set-CMEnrollmentPoint

Set-CMEnrollmentPoint

Sets an enrollment point in System Center 2012 Configuration Manager.

Syntax

Parameter Set: SetByName

```
Set-CMEnrollmentPoint -SiteCode <String> -SiteSystemServerName <String> [-UseComputerAccount] [-UserName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetByValue

```
Set-CMEnrollmentPoint -InputObject <IResultObject> [-UseComputerAccount] [-UserName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Set-CMEnrollmentPoint** cmdlet sets an enrollment point in Microsoft System Center 2012 Configuration Manager. An enrollment point is a site system role that uses public key infrastructure (PKI) certificates to complete mobile device enrollment and to provision Intel AMT-based computers.

Parameters

-InputObject<IResultObject>

Specifies an input object. To obtain an input object, use the **Get-CMEnrollmentPoint** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteCode<String>

Specifies the site code for a System Center 2012 Configuration Manager site.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteSystemServerName<String>

Specifies the fully qualified domain name (FQDN) of the server that hosts the site system role.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UseComputerAccount

Indicates that you use the computer account to connect to the Configuration Manager database.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UserName<String>

Specifies a user account that the enrollment point uses to connect to the Configuration Manager database.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Set an enrollment point

The command sets an enrollment point, and specifies an account name to use to connect to the Configuration Manager database.

```
PS C:\> Set-CMEnrollmentPoint -SiteSystemServerName "CM-Contoso.Contoso.Com" -SiteCode "CM2" -UserName "Contoso\ElisaDaugherty"
```

Example 2: Set an enrollment point with the computer account

The command sets an enrollment point by specifying the site system server and site code, and uses the computer account to connect to the Configuration Manager database.

```
PS C:\> Set-CMEnrollmentPoint -SiteSystemServerName "CM-Contoso.Contoso.Com" -SiteCode "CM2" -UseComputerAccount
```

Example 3: Set an enrollment point by using an input object

The first command uses the **Get-CMEnrollmentPoint** cmdlet to get an enrollment point, and stores the result in the \$Ep variable.

The second command sets an enrollment point for a server by using the input object stored in the \$Ep variable.

```
PS C:\> $Ep = Get-CMEnrollmentPoint -SiteSystemServerName "CM-Contoso.Contoso.Com" -SiteCode "CM2"
PS C:\> Set-CMEnrollmentPoint -InputObject $Ep -UserName "Contoso\ElisaDaugherty"
```

Related topics

[Add-CMEnrollmentPoint](#)

[Get-CMEnrollmentPoint](#)

[Remove-CMEnrollmentPoint](#)

Set-CMExchangeServer

Set-CMExchangeServer

Changes settings for an Exchange server.

Syntax

Parameter Set: SetBySiteCode

```
Set-CMExchangeServer -ServerAddress <String> -SiteCode <String> [-AccessLevel {Allow | Block | Quarantine} ] [-AccessRuleString <String[]> ] [-ActiveDirectoryContainer <String[]> ] [-AllowExternalDeviceManagement <Boolean> ] [-ApplicationSetting <Dictionary<String><String>> ] [-DeltaSyncInterval <Int32> ] [-EmailAddress <String[]> ] [-EmailManagementSetting <Dictionary<String><String>> ] [-EnableAccessRule <Boolean> ] [-ExchangeClientAccessServer <Dictionary[]<String><String>> ] [-FindAll] [-FullSyncSchedule <IResultObject> ] [-GeneralSetting <Dictionary<String><String>> ] [-IsHosted] [-MaximumInactiveDay <Int32> ] [-NewServerAddress <String> ] [-PasswordSetting <Dictionary<String><String>> ] [-SecuritySetting <Dictionary<String><String>> ] [-UserName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Set-CMExchangeServer** cmdlet changes settings for a Microsoft Exchange Server.

System Center 2012 Configuration Manager works with Exchange Server to manage mobile devices that cannot run System Center 2012 Configuration Manager clients.

Parameters

-AccessLevel<AccessLevelType>

Specifies the type of access for the mobile devices. Access level applies to a mobile device that is not managed by a rule. Valid values are:

- Allow
- Block
- Quarantine

The acceptable values for this parameter are:

Allow	
Block	

Quarantine	
------------	--

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AccessRuleString<String[]>

Specifies an array of access rules. Access rules control whether a mobile device can access an Exchange server. Access rules apply to a family of devices.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ActiveDirectoryContainer<String[]>

Specifies an array of names of Active Directory containers. When this parameter appears, the Exchange Server connector searches for the device only in the Active Directory containers.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AllowExternalDeviceManagement<Boolean>

Indicates whether an external device management program can manage the device.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ApplicationSetting<Dictionary<String><String>>

Specifies application settings, such as allow or deny the installation of applications. For each dictionary entry in the array, specify the setting name as the key the configuration as the value. Valid values are: AllowUnsignedApplications, AllowUnsignedInstallationPackages, or Block a specific application.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeltaSyncInterval<Int32>

Specifies the interval, in minutes, at which the Exchange Server connector runs delta discovery. Delta discovery looks for new devices or changes to existing devices.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EmailAddress<String[]>

Specifies an array of email addresses.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EmailManagementSetting<Dictionary<String><String>>

Specifies email management settings, such as synchronization schedule, message format, and size of attachments. For each dictionary entry in the array, specify the setting name as the key and the configuration as the value.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnableAccessRule<Boolean>

Indicates whether to enable an access rule.

Aliases	none
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ExchangeClientAccessServer<Dictionary[]<String><String>>

Specifies Exchange Client Access servers, as key-value pairs.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-FindAll

Indicates that the discovery process find all mobile devices in an Exchange organization.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-FullSyncSchedule<IResultObject>

Specifies a result object that schedules the full discovery time for an Exchange Server connector.

Aliases	none
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-GeneralSetting<Dictionary<String><String>>

Specifies general settings for mobile devices that use the Exchange Server Connector. Settings you can specify for this parameter include:

- RequireManualSyncWhenRoaming
- RequireStorageCardEncryption
- UnapprovedInROMApplicationList
- DevicePolicyRefreshInterval
- MaxInactivityTimeDeviceLock

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-IsHosted

Indicates that the Exchange Server connector configuration is for a hosted or on-premise Exchange Server.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-MaximumInactiveDay<Int32>

Specifies the interval between times that the Exchange Server connector runs device discovery. The cmdlet checks the last sync time of the devices that Exchange Server manages.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-NewServerAddress<String>

Specifies a new server address for an Exchange server.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PasswordSetting<Dictionary<String><String>>

Specifies general password settings. Settings you can specify for this parameter include:

- AlphanumericDevicePasswordRequired
- DevicePasswordEnabled
- DevicePasswordExpiration
- DevicePasswordHistory
- MaxDevicePasswordFailedAttempts
- PasswordRecoveryEnabled
- MinDevicePasswordComplexCharacters
- MinDevicePasswordLength
- AlphanumericDevicePasswordRequired
- AllowSimpleDevicePassword

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SecuritySetting<Dictionary<String><String>>

Specifies a dictionary of security settings. Settings you can specify for this parameter include:

- AllowBluetooth
- AllowBrowser
- AllowCamera
- AllowDesktopSync
- AllowInternetSharing
- AllowIrDA
- AllowNonProvisionableDevices
- AllowRemoteDesktop
- AllowStorageCard
- AllowTextMessaging
- AllowWiFi

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ServerAddress<String>

Specifies the address of the Exchange Server for which the cmdlet configures the Exchange Server connector.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteCode<String>

Specifies the Exchange Server by using a site code.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UserName<String>

Specifies the user name that the connector uses to connect to the Exchange Server.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Change settings for an Exchange server

The first command uses the **New-CMExchangeServerConnectorGeneralSetting** cmdlet to add new settings to an Exchange Server connector in Configuration Manager, and stores the settings in the \$Gs variable.

The second command uses the **New-CMExchangeServerConnectorPasswordSetting** cmdlet adds new password settings to an Exchange Server connector in Configuration Manager, and stores the password settings in the \$Ps variable.

The third command uses the **New-CMExchangeServerConnectorEmailManagementSetting** cmdlet creates a set of e-mail management settings for a mobile device that uses an Exchange Server connector, and stores the password settings in the \$Em variable.

The fourth command uses the **New-CMExchangeServerConnectorSecuritySetting** cmdlet configures security options for an Exchange Server connector in Configuration Manager, and security settings in the \$Ss variable.

The fifth command uses the **New-CMExchangeServerConnectorApplicationSetting** cmdlet creates application-related settings for a mobile device that uses an Exchange Server connector, and stores the application settings in the \$As variable.

The final command changes settings for an Exchange Server for the Configuration Manager site that has the site code CM2. The command specifies the general settings for the Exchange Server connector stored in \$Gs. The command specifies password settings for the Exchange Server connector stored in \$Ps. The command specifies a set of e-mail management settings for the Exchange Server connector stored in \$Em. The command specifies the security options for the Exchange Server connector stored in \$Ss. The command specifies application-related settings for a mobile device stored in \$As.

```
PS C:\> $Gs= New-CMExchangeServerConnectorGeneralSetting -AllowInternetShare $True -
AllowDesktopSync $True -AllowNonProvision $True -RefreshInterval 4
  PS C:\> $Ps= New-CMExchangeServerConnectorPasswordSetting -PasswordEnabled $True -
MinimumPasswordLength 8 -PasswordExpiration 51 -PasswordHistory 21 -WipeAfterFailedAttempt 6
-MaximumIdleTimeMinutes 41 -PasswordComplexity
  PS C:\> $Em = New-CMExchangeServerConnectorEmailManagementSetting -ConsumerEmail $True -
MaximumEmailAge OneDay -MaximumCalendarAge ThreeMonths -PushWhenRoaming $True -
AllowHtmlEmail $True -EmailAttachmentPolicy $True -MaximumSizeTextEmail 401 -
MaximumSizeHtmlEmail 402 -MaximumSizeAttachment 24
  PS C:\> $Ss = New-CMExchangeServerConnectorSecuritySetting -RemoteDesktop $True -
StorageCard $True -Camera $True -Bluetooth $False -WiFiConnection HandsfreeOnly -Infra
$False -Browser $False -StorageCardEncrypt $False -FileEncrypt $False -TextMessage $False
  PS C:\> $As= New-CMExchangeServerConnectorApplicationSetting -UnsignedInstall $True -
UnsignedApplication $False -BlockedApplication "App01","App02"
  PS C:\> Set-CMExchangeServer -SiteCode "CM2" -ServerAddress
"http://www.contoso.com/powershell" -NewServerAddress "www.fabrikam.com" -UserName
"ElisaDaugherty@contoso.com" -DeltaSyncInterval 124 -MaximumInactiveDay 26 -FindAll -
AllowExternalDeviceManagement $False -EnableAccessRule $True -AccessLevel Allow -
EmailAddress "EvanNarvaez@fabrikam.com","DavidChew@contosco.com" -GeneralSetting $Gs -
PasswordSetting $Ps -EmailManagementSetting $Em -SecuritySetting $Ss -ApplicationSetting $As
```

Related topics

[Get-CMExchangeServer](#)

[New-CMExchangeServer](#)

[Remove-CMExchangeServer](#)

[Sync-CMExchangeServer](#)

[New-CMExchangeServerConnectorSecuritySetting](#)

[New-CMExchangeServerConnectorApplicationSetting](#)

[New-CMExchangeServerConnectorEmailManagementSetting](#)

[New-CMExchangeServerConnectorGeneralSetting](#)

Set-CMFallbackStatusPoint

Set-CMFallbackStatusPoint

Changes the throttle interval or the message count for a Configuration Manager fallback status point.

Syntax

Parameter Set: SetByName

```
Set-CMFallbackStatusPoint -SiteCode <String> -SiteSystemServerName <String> [-StateMessagesCount <Int32> ] [-ThrottleMinutesInterval <Int32> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetByValue

```
Set-CMFallbackStatusPoint -InputObject <IResultObject> [-StateMessagesCount <Int32> ] [-ThrottleMinutesInterval <Int32> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Set-CMFallbackStatusPoint** cmdlet changes the throttle interval or the message count for a fallback status point. A fallback status point is a site system role. You can specify the site system name and site code for a fallback status point or use the **Get-CMFallbackStatusPoint** cmdlet to obtain a fallback status point object.

Microsoft System Center 2012 Configuration Manager can use one or more fallback status points to collect state messages for a site and send them to a server that is running System Center 2012 Configuration Manager. Throttling prevents the fallback status point from sending too many messages together, which can affect performance. You can use the *StateMessagesCount* and *ThrottleMinutesInterval* parameters to limit how many messages a fallback status point sends during a defined period.

Parameters

-InputObject<IResultObject>

Specifies a fallback status point role. To obtain a fallback status point role, use the **Get-CMFallbackStatusPoint** cmdlet.

Aliases	none
Required?	true

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteCode<String>

Specifies the site code for a fallback status point.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteSystemServerName<String>

Specifies the site system name for a fallback status point.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-StateMessagesCount<Int32>

Specifies the number of state messages that the fallback status point can send to Configuration Manager within a throttle interval. The default value is 10,000.

You can change the throttle interval by specifying the *ThrottleMinutesInterval* parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ThrottleMinutesInterval<Int32>

Specifies a throttle interval, in minutes, for a fallback status point. Configuration Manager processes a limited number of state messages during this period. The default value is 60 minutes.

You can change the limit to the number of messages by specifying the *StateMessagesCount* parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Change message and threshold settings for a fallback status point

The first command gets a fallback status point for the site that has the site code CM1 and the system name Server21.West01.Contoso.com and stores that object in the \$CMFSP variable.

The second command sets the count of state messages to 1,000 and the throttle interval to 60 minutes for the object stored in \$CMFSP.

```
PS C:\> $CMFSP = Get-CMFallbackStatusPoint -SiteCode "CM1" -SiteSystemServerName  
"Server21.West01.Contoso.com"  
PS C:\> Set-CMFallbackStatusPoint -InputObject $CMFSP -StateMessagesCount 1000 -  
ThrottleMinutesInterval 60
```

Example 2: Change message and threshold settings

This command sets the count of state messages to 1,000 and the throttle interval to 60 minutes for the fallback status point for the site that has the site code CM1 and the system name Server21.West01.Contoso.com.

```
PS C:\> Set-CMFallbackStatusPoint -SiteCode "CM1" -SiteSystemServerName  
"Server21.West01.Contoso.com" -StateMessagesCount 1000 -ThrottleMinutesInterval 60
```

Related topics

[Add-CMFallbackStatusPoint](#)

[Get-CMFallbackStatusPoint](#)

[Remove-CMFallbackStatusPoint](#)



Set-CMFileReplicationRoute

Set-CMFileReplicationRoute

Changes settings for a file replication route in Configuration Manager.

Syntax

Parameter Set: SetFileReplicationRouteByLimited

```
Set-CMFileReplicationRoute -DestinationSiteCode <String> -Limited -SourceSiteCode <String> [-FileReplicationAccountName <String> ] [-LimitAvailableBandwidthPercentage <Int32> ] [-LimitedTimePeriodEnd <Int32> ] [-LimitedTimePeriodStart <Int32> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetFileReplicationRouteByPulseMode

```
Set-CMFileReplicationRoute -DestinationSiteCode <String> -PulseMode -SourceSiteCode <String> [-DataBlockSizeKB <Int32> ] [-DelayBetweenDataBlocksSeconds <Int32> ] [-FileReplicationAccountName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetFileReplicationRouteBySchedule

```
Set-CMFileReplicationRoute -ControlNetworkLoadSchedule -DestinationSiteCode <String> -SourceSiteCode <String> [-AvailabilityLevel {All | Closed | High | MediumHigh} ] [-DaysOfWeek {Friday | Monday | Saturday | Sunday | Thursday | Tuesday | Wednesday} ] [-TimePeriodEnd <Int32> ] [-TimePeriodStart <Int32> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetFileReplicationRouteByUnlimited

```
Set-CMFileReplicationRoute -DestinationSiteCode <String> -SourceSiteCode <String> -Unlimited [-FileReplicationAccountName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Set-CMFileReplicationRoute** cmdlet changes settings for a file replication route from Microsoft System Center 2012 Configuration Manager. System Center 2012 Configuration Manager uses file replication routes to transfer file-based data between sites in a hierarchy. Each file replication route identifies a destination site to which file-based data can transfer.

File replication routes were known as addresses in versions of Configuration Manager before System Center 2012 Configuration Manager. The functionality of file replication routes is the same as that of addresses in earlier versions.

Parameters

-AvailabilityLevel<AvailabilityLevel>

Specifies a value that indicates the priorities for which the scheduled restriction allows. The system allows all priorities, no priorities, high priority only or high and medium priority. Valid values are:

- All
- Closed
- High
- MediumHigh

The acceptable values for this parameter are:

All	
Closed	
High	
MediumHigh	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ControlNetworkLoadSchedule

Indicates that scheduled replication controls network load.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DataBlockSizeKB<Int32>

Specifies a data block size, in kilobytes.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DaysOfWeek<DaysOfWeek[]>

Specifies an array of values that indicate when the file replication runs for this route. Valid values are:

- Friday
- Saturday
- Sunday
- Monday
- Tuesday
- Wednesday
- Thursday

The acceptable values for this parameter are:

Friday	
Monday	
Saturday	
Sunday	
Thursday	
Tuesday	
Wednesday	

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DelayBetweenDataBlocksSeconds<Int32>

Specifies the delay between transmission of data blocks.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DestinationSiteCode<String>

Specifies the destination site in the file replication route that you change by using a site code.

Aliases	DesSiteCode
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-FileReplicationAccountName<String>

Specifies the account that Configuration Manager uses to install a site on the specified server and maintain communications between the site and other sites. This account must have local administrative credentials.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-LimitAvailableBandwidthPercentage<Int32>

Specifies the percentage of available bandwidth to restrict on a network replication route.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Limited

Indicates that bandwidth for a file replication route is limited.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-LimitedTimePeriodEnd<Int32>

Specifies when the time period of limited bandwidth ends.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-LimitedTimePeriodStart<Int32>

Specifies when the time period of limited bandwidth starts.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PulseMode

Indicates that file replication uses data block size and delays between transmissions. Use this parameter when you have low network bandwidth between sites.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SourceSiteCode<String>

Specifies the source site in the file replication route that you change by using a site code.

Aliases	SiteCode
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-TimePeriodEnd<Int32>

Specifies the end time for file replication.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-TimePeriodStart<Int32>

Specifies the start time for file replication.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Unlimited

Indicates that bandwidth for a file replication route is unlimited.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Specify a file replication route by using a replication account name

This command specifies a file replication route between the source site named CM2 and the destination site named SS2. It uses the user account name 11\12 for file replication.

```
PS C:\> Set-CMFileReplicationRoute -SourceSiteCode "CM2" -DestinationSiteCode "SS2" -  
FileReplicationAccountName "11\12" -Unlimited
```

Example 2: Specify a file replication route by using a source and destination site names

This command specifies a file replication route between the source site named CM2 and the destination site named SS2. It schedules file replication for Fridays and Sundays.

```
PS C:\> Set-CMFileReplicationRoute -SourceSiteCode "CM2" -DestinationSiteCode "SS2" -  
ControlNetworkLoadSchedule -DaysOfWeek Friday, Sunday -AvailabilityLevel All
```

Related topics

[Get-CMFileReplicationRoute](#)

[New-CMFileReplicationRoute](#)

[Remove-CMFileReplicationRoute](#)

Set-CMGlobalCondition

Set-CMGlobalCondition

Modifies settings for a Configuration Manager global condition.

Syntax

Parameter Set: SetADQuery

```
Set-CMGlobalCondition -Name <String> [-DistinguishedName <String> ] [-LdapFilter <String> ]  
[-LdapPrefix <String> ] [-Property <String> ] [-SearchScope {Base | OneLevel | Subtree} ] [-  
Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetAssembly

```
Set-CMGlobalCondition -Name <String> [-AssemblyName <String> ] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SetFileSystem

```
Set-CMGlobalCondition -Name <String> [-FileOrFolderName <String> ] [-IncludeSubfolders] [-  
Is64Bit] [-Path <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetFileSystemFile

```
Set-CMGlobalCondition -Name <String> [-FilePath <String> ] [-IncludeSubfolders] [-Is64Bit]  
[-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetGeneral

```
Set-CMGlobalCondition -Name <String> [-Description <String> ] [-NewName <String> ] [-  
Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetIisMetabase

```
Set-CMGlobalCondition -Name <String> [-MetabasePath <String> ] [-PropertyId <String> ] [-  
Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetOmaUri

```
Set-CMGlobalCondition -Name <String> [-OmaUri <String> ] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SetRegistryKey

```
Set-CMGlobalCondition -Name <String> [-Is64Bit] [-KeyName <String> ] [-RegistryHive  
{ClassesRoot | CurrentConfig | CurrentUser | LocalMachine | Users} ] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SetRegistryValue

```
Set-CMGlobalCondition -Name <String> [-Is64Bit] [-KeyName <String> ] [-RegistryHive  
{ClassesRoot | CurrentConfig | CurrentUser | LocalMachine | Users} ] [-ValueName <String> ]  
[-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetScript

```
Set-CMGlobalCondition -Name <String> [-FilePath <String> ] [-ScriptLanguage {PowerShell |
VBScript | JScript | ShellScript} ] [-Use32BitHost] [-UseLoggedInUserCredentials] [-Confirm]
[-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetSecurityScopeByName

```
Set-CMGlobalCondition -Name <String> -SecurityScopeAction {AddMembership | RemoveMembership}
-SecurityScopeName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetSqlQueryAllInstances

```
Set-CMGlobalCondition -Name <String> [-Column <String> ] [-Database <String> ] [-FilePath
<String> ] [-UseAllInstances] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetSqlQueryDefaultInstance

```
Set-CMGlobalCondition -Name <String> [-Column <String> ] [-Database <String> ] [-FilePath
<String> ] [-UseDefaultInstance] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetSqlQuerySpecificInstance

```
Set-CMGlobalCondition -Name <String> [-Column <String> ] [-Database <String> ] [-FilePath
<String> ] [-InstanceName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetWqlQuery

```
Set-CMGlobalCondition -Name <String> [-Class <String> ] [-Namespace <String> ] [-Property
<String> ] [-WhereClause <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetXPathQuery

```
Set-CMGlobalCondition -Name <String> [-FilePath <String> ] [-IncludeSubfolders] [-Is64Bit]
[-XmlFilePath <String> ] [-XmlNamespace <String[]> ] [-Confirm] [-WhatIf] [
<CommonParameters>]
```

Detailed Description

The **Set-CMGlobalCondition** cmdlet modifies settings for a global condition. You can add or remove a security scope for a global condition. You can specify a global condition by name or ID, or use the **Get-CMGlobalCondition** cmdlet to obtain a global condition object.

Microsoft System Center 2012 Configuration Manager uses global conditions to represent business or technical conditions. Global conditions specify how to provide and deploy applications to client devices. Each global condition must have at least one security scope.

Parameters

-AssemblyName<String>

Specifies the name of an assembly for which to search. An assembly name must be registered in the global assembly cache.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Class<String>

Specifies a Windows Management Instrumentation (WMI) class used to build a WMI Query Language (WQL) query. The query assesses compliance on client computers.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Column<String>

Specifies the column name used to assess the compliance of the global condition.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Database<String>

Specifies the name of a database. The SQL query runs on this database.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Description<String>

Specifies a description for the global condition.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DistinguishedName<String>

Specifies the distinguished name of the Active Directory Domain Services (AD DS) object to assess for compliance on client computers.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-FileOrFolderName<String>

Specifies the name of a file or folder. Specify the *IsFolder* parameter to search for a folder.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-FilePath<String>

Specifies a file path for the file that the condition assesses for compliance.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-IncludeSubfolders

Indicates that the global condition searches in subfolders.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InstanceName<String>

Specifies the name of a database instance that the global condition searches. To search the default instance, specify the *UseDefaultInstance* parameter. To search all instances, specify the *UseAllInstances* parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Is64Bit

Indicates that the global condition searches the 64-bit system file location in addition to the 32-bit system file location.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-KeyName<String>

Specifies the registry key name for which to search. Use the format *key\subkey*.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-LdapFilter<String>

Specifies an LDAP filter to refine the results from the AD DS query to assess compliance on client computers.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-LdapPrefix<String>

Specifies a valid Lightweight Directory Access Protocol (LDAP) prefix for the AD DS query that assesses compliance on client computers. Valid values are: LDAP:// or GC://.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-MetabasePath<String>

Specifies the path to the metabase file for Internet Information Services (IIS).

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-Name<String>

Specifies the name of the global conditions. This value corresponds to the **LocalizedDisplayName** property of a global condition object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Namespace<String>

Specifies a namespace from a WMI repository. The default value is Root\cimv2.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-NewName<String>

Specifies a new name for the global condition.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OmaUri<String>

Specifies a Uniform Resource Indicator (URI) that points to device-specific parameters for an Open Mobile Alliance (OMA) device.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Path<String>

Specifies the path for an OMA URI.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Property<String>

Specifies the property of the AD DS object used to assess compliance on client computers.

Aliases	none
Required?	false
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PropertyId<String>

Specifies the property of AD DS that Configuration Manager uses to determine client compliance.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RegistryHive<RegistryRootKey>

Specifies the root key in the registry that identifies the registry hive that you search. WMI uses the registry hive to return, set, and change the values of registry keys. Valid values are: ClassesRoot, CurrentConfig, CurrentUser, LocalMachine, and Users.

The acceptable values for this parameter are:

ClassesRoot	
CurrentConfig	
CurrentUser	
LocalMachine	
Users	

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ScriptLanguage<ScriptingLanguage>

Specifies a scripting language to use. Valid values are: PowerShell, VBScript, and JScript.

The acceptable values for this parameter are:

PowerShell	
VBScript	
JScript	
ShellScript	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SearchScope<SearchScope>

Specifies the search scope in AD DS. Valid values are: Base, OneLevel, and Subtree.

The acceptable values for this parameter are:

Base	
OneLevel	
Subtree	

Aliases	none
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SecurityScopeAction<SecurityScopeActionType>

Specifies an action for a security scope. Valid values are: AddMembership and RemoveMembership.

The acceptable values for this parameter are:

AddMembership	
RemoveMembership	

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SecurityScopeName<String>

Specifies the name of a security scope. A security scope name can be Default or the name of a custom security scope.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Use32BitHost

Indicates that the file or folder is associated with a 64-bit application.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UseAllInstances

Indicates that the global condition searches all database instances. To search a named instance, specify the *InstanceName* parameter. To search the default instance, specify the *UseDefaultInstance* parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UseDefaultInstance

Indicates that the global condition searches the default database instance. To search a named instance, specify the *InstanceName* parameter. To search all instances, specify the *UseAllInstances* parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-UseLoggedOnUserCredentials

Indicates that the script runs on client computers by using the logged on user credentials.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ValueName<String>

Specifies the value to be contained in the specified registry key.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-WhereClause<String>

Specifies a WQL query WHERE clause to apply to the specified namespace, class, and property on client computers.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-XmlFilePath<String>

Specifies a file that contains the XML query to use to assess compliance on client computers.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-XmlNamespace<String[]>

Specifies an array of valid, full XML path language (XPath) queries to use to assess compliance on client computers.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Add a security scope

This command adds the security scope named Scope22 to the global condition named CPU speed.

```
PS C:\> Set-CMGlobalCondition -Name "CPU speed" -SecurityScopeAction AddMembership -
SecurityScopeName "Scope22"
```

Example 2: Remove a security scope by using a variable

The first command uses the **Get-CMGlobalCondition** cmdlet to get the global condition named CPU speed and store it in the \$CMGC variable.

The second command removes the security scope named Scope22 from the global condition stored in the \$CMGC variable.

```
PS C:\> $CMGC = Get-CMGlobalCondition -Name "CPU speed"
PS C:\> Set-CMGlobalCondition -InputObject $CMGC -SecurityScopeAction RemoveMembership -
SecurityScopeName "Scope22"
```

Related topics

[Get-CMGlobalCondition](#)

[New-CMGlobalCondition](#)

[Remove-CMGlobalCondition](#)

Set-CMHardwareRequirement

Set-CMHardwareRequirement

Changes Configuration Manager hardware requirement settings for a product.

Syntax

Parameter Set: SetByName

```
Set-CMHardwareRequirement -Product <String> [-MinCpu <Int32> ] [-MinDiskFree <Int64> ] [-MinDiskSize <Int64> ] [-MinRam <Int64> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetByValueMandatory

```
Set-CMHardwareRequirement -InputObject <IResultObject> [-MinCpu <Int32> ] [-MinDiskFree <Int64> ] [-MinDiskSize <Int64> ] [-MinRam <Int64> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Set-CMHardwareRequirement** cmdlet changes settings for hardware requirements for software products.

Microsoft System Center 2012 Configuration Manager manages Asset Intelligence information, including hardware requirements, for different software products. You can add, modify, or delete your own hardware requirements, but you cannot change built-in hardware requirements.

You can use this cmdlet to modify the minimum requirements associated with a software product or change the name that System Center 2012 Configuration Manager uses for a product. You can specify a product by name or obtain a product by using the **Get-CMHardwareRequirement** cmdlet.

Parameters

-InputObject<IResultObject>

Specifies a hardware requirement object. To obtain a hardware requirement object, use the **Get-CMHardwareRequirement** cmdlet.

Aliases	none
Required?	true
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-MinCpu<Int32>

Specifies a minimum CPU speed, in megahertz (MHz), required for a software product.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-MinDiskFree<Int64>

Specifies a minimum amount of available disk memory, in kilobytes (KB), required for a software product.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-MinDiskSize<Int64>

Specifies a minimum disk size, in kilobytes, required for a software product.

Aliases	none
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-MinRam<Int64>

Specifies a minimum amount of random access memory (RAM), in kilobytes, required for a software product.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Product<String>

Specifies the name of a software product name.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Change minimum RAM value

This command sets the minimum RAM value for a specified product.

```
PS C:\> Set-CMHardwareRequirement -Product "Accounts Program" -MinRam 161072
```

Example 2: Change minimum disk size value for a hardware requirements object

The first command gets the hardware requirements object for Accounts Program and stores it in the \$CMHR variable.

The second command changes the minimum disk size for the object stored in \$CMHR.

```
PS C:\> $CMHR = Get-CMHardwareRequirement -Product "Accounts Program"
PS C:\> Set-CMHardwareRequirement -InputObject $CMHR -MinDiskSize 1600000
```

Related topics

[Get-CMHardwareRequirement](#)

[New-CMHardwareRequirement](#)

[Remove-CMHardwareRequirement](#)

Set-CMHierarchySetting

Set-CMHierarchySetting

Syntax

```
Parameter Set: SetHierarchySetting
Set-CMHierarchySetting [-AllowPrestage <Boolean> ] [-ApprovalMethod
{AutomaticallyApproveAllComputers | AutomaticallyApproveComputersInTrustedDomains |
ManuallyApproveEachComputer} ] [-AutomaticallyResolveConflictingRecord <Boolean> ] [-
AutomaticallyUpgradeDays <Int32> ] [-EnableProgram <Boolean> ] [-FallbackSiteCode <String> ]
[-Force] [-UseFallbackSite <Boolean> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Set-CMHierarchySetting** cmdlet

Parameters

-AllowPrestage<Boolean>

Indicates whether to allow prestaging.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ApprovalMethod<ApprovalMethodType>

Specifies an approval method. Valid values are:

- AutomaticallyApproveAllComputers
- AutomaticallyApproveComputersInTrustedDomains

-- ManuallyApproveEachComputer

The acceptable values for this parameter are:

AutomaticallyApproveAllComputers	
AutomaticallyApproveComputersInTrustedDomains	
ManuallyApproveEachComputer	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AutomaticallyResolveConflictingRecord<Boolean>

Indicates whether to automatically resolve record conflicts.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AutomaticallyUpgradeDays<Int32>

Specifies the number of days before an automatic upgrade occurs.

Aliases	none
Required?	false
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnableProgram<Boolean>

Indicates whether to enable a program.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-FallbackSiteCode<String>

Specifies the site code for a fallback site.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UseFallbackSite<Boolean>

Indicates whether to use a fallback site.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Modify the hierarchy setting

This command uses the **Set-CMHierarchySetting** cmdlet to modify the hierarchy setting. The command specifies the value *AutomaticallyApproveAllComputers* for the *ApprovalMethod* parameter, and also specifies the *AllowPrestage* parameter.

```
PS C:\> Set-CMHierarchySetting -AllowPrestage -ApprovalMethod  
AutomaticallyApproveAllComputers
```

Set-CMMaintenanceWindow

Set-CMMaintenanceWindow

Modifies a maintenance window.

Syntax

Parameter Set: ByScheduleMandatory

```
Set-CMMaintenanceWindow [-CollectionID] <String> -Name <String> [-Schedule <IResultObject> ]  
[-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByScheduleWindowMandatory

```
Set-CMMaintenanceWindow [-CollectionID] <String> -ScheduleWindow <IResultObject> [-Confirm]  
[-WhatIf] [ <CommonParameters>]
```

Parameter Set: SoftwareUpdateOnly

```
Set-CMMaintenanceWindow [-CollectionID] <String> -ApplyToSoftwareUpdateOnly -Name <String>  
[-Schedule <IResultObject> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: TaskSequenceOnly

```
Set-CMMaintenanceWindow [-CollectionID] <String> -ApplyToTaskSequenceOnly -Name <String> [-  
Schedule <IResultObject> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Set-CMMaintenanceWindow** cmdlet modifies a maintenance window associated with a collection. Maintenance windows are periods of time reserved for write operations such as applying software updates, installing software, or configuring computer settings.

Parameters

-ApplyToSoftwareUpdateOnly

Indicates that the maintenance window is used to apply software updates only.

Aliases	none
Required?	true
Position?	named

Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

-ApplyToTaskSequenceOnly

Indicates that the maintenance window is used to apply task sequences only.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

-CollectionID<String>

Specifies the ID of the collection that the maintenance window applies to.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies the name of the maintenance window.

Aliases	none
Required?	true
Position?	named

Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

-Schedule<IResultObject>

Specifies a **CMSchedule** object. The schedule specifies when the maintenance window occurs. To create a **CMSchedule** object, use the **New-CMSchedule** cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

-ScheduleWindow<IResultObject>

Specifies a maintenance window object. To obtain a maintenance window object, use the **Get-CMMaintenanceWindow** cmdlet.

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](#).

Related topics

[Get-CMMaintenanceWindow](#)

[New-CMMaintenanceWindow](#)

[Remove-CMMaintenanceWindow](#)

[New-CMSchedule](#)

Set-CMManagementPoint

Set-CMManagementPoint

Changes settings for a management point in Configuration Manager.

Syntax

Parameter Set: SetByName

```
Set-CMManagementPoint -SiteCode <String> -SiteSystemServerName <String> [-AllowDevice <Boolean> ] [-ClientConnectionType {Internet | InternetAndIntranet | Intranet} ] [-DatabaseName <String> ] [-EnableSsl <Boolean> ] [-GenerateAlert <Boolean> ] [-SqlServerFqdnName <String> ] [-SqlServerInstanceName <String> ] [-UseComputerAccount] [-UserName <String> ] [-UseSiteDatabase <Boolean> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetByValue

```
Set-CMManagementPoint -InputObject <IResultObject> [-AllowDevice <Boolean> ] [-ClientConnectionType {Internet | InternetAndIntranet | Intranet} ] [-DatabaseName <String> ] [-EnableSsl <Boolean> ] [-GenerateAlert <Boolean> ] [-SqlServerFqdnName <String> ] [-SqlServerInstanceName <String> ] [-UseComputerAccount] [-UserName <String> ] [-UseSiteDatabase <Boolean> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Set-CMManagementPoint** cmdlet changes settings for a management point in Microsoft System Center 2012 Configuration Manager. A management point is a System Center 2012 Configuration Manager site that provides policy and service information to clients and receives configuration data from clients.

Parameters

-AllowDevice<Boolean>

Indicates whether the management point supports device clients.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ClientConnectionType<ClientConnectionTypes>

Specifies the type of the client connection. Valid values are:

- Internet
- InternetAndIntranet
- Intranet

The acceptable values for this parameter are:

Internet	
InternetAndIntranet	
Intranet	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DatabaseName<String>

Specifies the name of the site database or site database replica that the management point uses to query for site database information.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-EnableSsl<Boolean>

Indicates whether to enable SSL.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-GenerateAlert<Boolean>

Indicates whether the management point generates health alerts.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies the management point for which you change values by using a management point object. To obtain a management point object, use the **Get-CMManagementPoint** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteCode<String>

Specifies the site code of the System Center 2012 Configuration Manager site that hosts the site system role.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteSystemServerName<String>

Specifies the name of the server that hosts the site system role.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SqlServerFqdnName<String>

Specifies the server name of the site database or site database replica that the management point uses to query for site database information. You must specify this parameter if Internet-based client systems communicate with the site system.

Aliases	none
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SqlServerInstanceName<String>

Specifies the name of the SQL Server instance that clients use to communicate with the site system.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UseComputerAccount

Indicates that the management point uses its own computer account instead of a domain user account to access site database information.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UserName<String>

Specifies a domain user account that the management point uses to access site information.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UseSiteDatabase<Boolean>

Indicates whether the management point queries a site database rather than a site database replica for information.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Change management point settings for site system and site code

This command changes settings for a management point in a System Center 2012 Configuration Manager installation. The command specifies the following information about the management point:

- The new management point appears on the site system named CMDEV-TEST02.TSQA.CONTOSO.COM.
- The site has code CM2.
- The management point queries a site database for information.
- The command enables SSL for the management point.

```
PS C:\> Set-CMManagementPoint -SiteSystemServerName "CMDEV-TEST02.TSQA.CORP.CONTOSO.COM" -  
SiteCode "CM2" -EnableSSL $False -UseSiteDatabase $True
```

Example 2: Change management point settings for site system and site code, connection type, and database name

This command changes settings for a management point in a System Center 2012 Configuration Manager installation. The command specifies the following information about the management point:

- The new management point appears on the site system named CMDEV-TEST02.TSQA.CONTOSO.COM. This name is also the fully qualified domain name for the SQL Server instance named MSSQLServer.
- The site has code CM2.
- The management point accepts connections from internet and intranet clients and from portable devices.
- The management point has the associated database name ContosoSQL01.

-- The command enables SSL for the management point.

```
PS C:\> Set-CMManagementPoint -SiteSystemServerName "CMDEV-TEST02.TSQA.CORP.CONTOSO.COM " -  
SiteCode "CM2" -ClientConnectionType InternetAndIntranet -AllowDevice $True -EnableSSL $True  
-GenerateAlert $false -UseSiteDatabase $False -SQLServerFqdnName "CMDEV-  
TEST02.TSQA.CORP.CONTOSO.COM" -SQLServerInstanceName "MSSQLServer" -DatabaseName  
"ContosoSQL01"
```

Related topics

[Add-CMManagementPoint](#)

[Get-CMManagementPoint](#)

[Remove-CMManagementPoint](#)

Set-CMManagementPointComponent

Set-CMManagementPointComponent

Sets a component for a management point in Configuration Manager.

Syntax

Parameter Set: SearchBySiteCodeMandatory

```
Set-CMManagementPointComponent -PublishDns <Boolean> -SiteCode <String> [-Confirm] [-WhatIf] [  
 <CommonParameters>]
```

Parameter Set: SearchByNameMandatory

```
Set-CMManagementPointComponent -Name <String> -PublishDns <Boolean> [-Confirm] [-WhatIf] [  
 <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Set-CMManagementPointComponent -InputObject <IResultObject> -PublishDns <Boolean> [-Confirm] [  
 -WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Set-CMManagementPointComponent** cmdlet sets a component for a management point in Microsoft System Center 2012 Configuration Manager.

Parameters

-InputObject<IResultObject>

Specifies an input object. To obtain an input object, use the **Get-CMManagementPointComponent** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies a name in Configuration Manager.

Aliases	SiteName
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PublishDns<Boolean>

Indicates whether to publish resource records for the management point component in the Domain Name System (DNS). Configuration Manager clients use DNS service location resource records (SRV RR) to find a management point in a site.

For Configuration Manager to publish management points to DNS, your DNS servers must have version 8.1.2 of BIND, or later. Your DNS servers must be configured for automatic updates and support service location resource records. The fully qualified domain names (FQDNs) of management points must have host entries in DNS.

You must assign clients to a specific site and configure the clients to use the site code with the domain suffix of their management point.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteCode<String>

Specifies a site code in Configuration Manager.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Set a management point component

The command uses the **Set-CMManagementPointComponent** cmdlet to set a management point component by using the *SiteCode* parameter. The command also sets the *PublishDNS* parameter to `$True` to publish the management point component in DNS.

```
PS C:\> Set-CMManagementPointComponent -SiteCode "CM1" -PublishDNS $True
```

Related topics

[Get-CMManagementPointComponent](#)

Set-CMMigrationExclusionList

Set-CMMigrationExclusionList

Edits the global exclusion list for migration jobs.

Syntax

Parameter Set: EditExclusionListByName

```
Set-CMMigrationExclusionList -Name <String[]> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Set-CMMigrationExclusionList** cmdlet edits the global exclusion list for migration jobs in Microsoft System Center 2012 Configuration Manager.

For a collection-based migration, you specify one or more collections to migrate. For each collection that you specify, the migration job automatically selects all related objects for migration. Objects on the exclusion list are available for migration, but System Center 2012 Configuration Manager does not automatically include these objects when you create a new collection-based migration job.

Parameters

-Name<String[]>

Specifies an array of objects that you exclude by default from collection-based migration jobs.

Aliases	EntityName
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Specify a migration exclusion list

This command adds the objects in the array ContosoUsersWest01 to the migration exclusion list.

```
PS C:\> Set-CMMigrationExclusionList -Name "ContosoUsersWest01"
```

Related topics

[Clear-CMMigrationData](#)

[Set-CMMigrationSource](#)

Set-CMMigrationSource

Set-CMMigrationSource

Specifies or changes settings for a migration source site in Configuration Manager.

Syntax

Parameter Set: SetMigrationSource

```
Set-CMMigrationSource -SourceSiteServerName <String> [-EnableDistributionPointSharing <Boolean> ] [-SmsProviderAccount <String> ] [-SqlServerAccount <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Set-CMMigrationSource** cmdlet specifies or changes settings for a migration source site in Microsoft System Center 2012 Configuration Manager. To enable migration of data to your System Center 2012 Configuration Manager environment, you must configure a supported System Center 2012 Configuration Manager source hierarchy and specify one or more source sites that contain data that you want to migrate.

By default, the top-level site of the hierarchy becomes a source site of the source hierarchy. If you migrate from a Microsoft System Center Configuration Manager 2007 hierarchy, you can configure additional source sites for migration. If you migrate from a System Center 2012 Configuration Manager hierarchy, you do not need to configure additional source sites because the System Center 2012 Configuration Manager shared database at the top of the source hierarchy contains all of the information that you can migrate.

Parameters

-EnableDistributionPointSharing<Boolean>

Indicates whether Configuration Manager shares distribution points between source and destination hierarchies.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SmsProviderAccount<String>

Specifies the name of the account that the top-level site in the destination hierarchy uses to connect to the SMS Provider and the site database of the source site. The top-level site uses these connections to retrieve objects and distribution points.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SourceSiteServerName<String>

Specifies the name of a server that contains a source site.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SqlServerAccount<String>

Specifies a SQL Server account that provides access to the top-level SQL Server database in the source hierarchy. The account must have read and run permissions for this database.

Aliases	none
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Specify a migration source

This command specifies the site server named cmdev-contoso01 as the source of migration data. It uses the user account for tsqa\pattifuller to provide access to the SQL Server database at the top of the source hierarchy and to connect to the SMS provider and source site database for the source site. The command also shares distribution points between the source and destination hierarchies.

```
PS C:\> Set-CMMigrationSource -SourceSiteServerName "cmdev-contoso01" -SmsProviderAccount  
"tsqa\pattifuller" -SqlServerAccount "tsqa\pattifuller" -EnableDistributionPointSharing  
$True
```

Related topics

[Set-CMMigrationExclusionList](#)

[Clear-CMMigrationData](#)

Set-CMOperatingSystemImage

Set-CMOperatingSystemImage

Changes configuration settings of operating system images.

Syntax

Parameter Set: SetById

```
Set-CMOperatingSystemImage -Id <String[]> [-Description <String> ] [-NewName <String> ] [-Path <String> ] [-SecuredScopeNames <String> ] [-Version <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetByName

```
Set-CMOperatingSystemImage -Name <String> [-Description <String> ] [-NewName <String> ] [-Path <String> ] [-SecuredScopeNames <String> ] [-Version <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetByValueMandatory

```
Set-CMOperatingSystemImage -InputObject <IResultObject> [-Description <String> ] [-NewName <String> ] [-Path <String> ] [-Version <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetSecurityScopeById

```
Set-CMOperatingSystemImage -Id <String[]> -SecurityScopeAction {AddMembership | RemoveMembership} -SecurityScopeName <String> [-SecuredScopeNames <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetSecurityScopeByName

```
Set-CMOperatingSystemImage -Name <String> -SecurityScopeAction {AddMembership | RemoveMembership} -SecurityScopeName <String> [-SecuredScopeNames <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetSecurityScopeByValue

```
Set-CMOperatingSystemImage -InputObject <IResultObject> -SecurityScopeAction {AddMembership | RemoveMembership} -SecurityScopeName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Set-CMOperatingSystemImage** cmdlet changes configuration settings of one or more operating system images in Microsoft System Center 2012 Configuration Manager. Operating system images are .wim format files and represent a compressed collection of reference files and folders that System Center 2012 Configuration Manager requires to successfully install and configure an operating system on a computer.

Parameters

-Description<String>

Specifies a description for the operating system image.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Id<String[]>

Specifies an array of IDs of operating system images.

Aliases	PackageId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a **CMOperatingSystemImage** object. To obtain a **CMOperatingSystemImage** object, use the **Get-CMOperatingSystemImage** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-Name<String>

Specifies the name of an operating system image.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-NewName<String>

Specifies the new name of an operating system image.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Path<String>

Specifies the network path to the operating system image source .wim file.

Aliases	PackageSourcePath
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-SecuredScopeNames<String>

Specifies names of security scopes. A security scope name can be Default or the name of a custom-created security scope.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SecurityScopeAction<SecurityScopeActionType>

Specifies an action for a security scope. Valid values are: AddMembership and RemoveMembership. The acceptable values for this parameter are:

AddMembership	
RemoveMembership	

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SecurityScopeName<String>

Specifies the name of a security scope. A security scope name can be Default or the name of a custom security scope.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Version<String>

Specifies the version of the operating system image.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Change settings for an operating system image by using an ID

This command changes configuration settings of the operating system image that has the ID Cm10004f. The command renames the operating system image, adds a version and description, and specifies the path to the installation source files of the operating system image.

```
PS C:\> Set-CMOperatingSystemImage -Id "Cm10004f" -NewName "Microsoft Windows 8 (x64)" -Version "I20C" -Description "Dept02 Sys Image" -Path "\\Contoso\Public\OSD\win8x64.wim"
```

Example 2: Add an operating system image to a security scope by using a name

This command adds membership to the security scope named SecScope02 for the operating system image named ImagePkg01.

```
PS C:\> Set-CMOperatingSystemImage -SecurityScopeAction AddMembership -SecurityScopeName "SecScope02" -Name "ImagePkg01"
```

Example 3: Remove an operating system image from a security scope

This command removes membership from the security scope named SecScope02 for the operating system image named ImagePkg01.

```
PS C:\> Set-CMOperatingSystemImage -SecurityScopeAction RemoveMembership -SecurityScopeName "SecScope02" -Name "ImagePkg01"
```

Related topics

[Get-CMOperatingSystemImage](#)

[New-CMOperatingSystemImage](#)

[Remove-CMOperatingSystemImage](#)

[Get-CMOperatingSystemImageUpdateSchedule](#)

Set-CMOperatingSystemImageUpdateSchedule

Set-CMOperatingSystemImageUpdateSchedule

Sets a schedule for an operating system image update in System Center 2012 Configuration Manager.

Syntax

Parameter Set: SetScheduleById

```
Set-CMOperatingSystemImageUpdateSchedule -Id <String> -SoftwareUpdate <IResultObject[]> [-ContinueOnError <Boolean> ] [-CustomSchedule <DateTime> ] [-UpdateDistributionPointsWithImage <Boolean> ] [-Utc <Boolean> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetScheduleByIdRunNow

```
Set-CMOperatingSystemImageUpdateSchedule -Id <String> -SoftwareUpdate <IResultObject[]> [-ContinueOnError <Boolean> ] [-RunNow] [-UpdateDistributionPointsWithImage <Boolean> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetScheduleByInputObject

```
Set-CMOperatingSystemImageUpdateSchedule -InputObject <IResultObject> -SoftwareUpdate <IResultObject[]> [-ContinueOnError <Boolean> ] [-CustomSchedule <DateTime> ] [-UpdateDistributionPointsWithImage <Boolean> ] [-Utc <Boolean> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetScheduleByInputObjectRunNow

```
Set-CMOperatingSystemImageUpdateSchedule -InputObject <IResultObject> -SoftwareUpdate <IResultObject[]> [-ContinueOnError <Boolean> ] [-RunNow] [-UpdateDistributionPointsWithImage <Boolean> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetScheduleByName

```
Set-CMOperatingSystemImageUpdateSchedule -Name <String> -SoftwareUpdate <IResultObject[]> [-ContinueOnError <Boolean> ] [-CustomSchedule <DateTime> ] [-UpdateDistributionPointsWithImage <Boolean> ] [-Utc <Boolean> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetScheduleByNameRunNow

```
Set-CMOperatingSystemImageUpdateSchedule -Name <String> -SoftwareUpdate <IResultObject[]> [-ContinueOnError <Boolean> ] [-RunNow] [-UpdateDistributionPointsWithImage <Boolean> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Set-CMOperatingSystemImageUpdateSchedule** cmdlet sets a schedule for updating an operating system image in Microsoft System Center 2012 Configuration Manager.

Parameters

-ContinueOnError<Boolean>

Indicates whether the update process continues if it encounters an error.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CustomSchedule<DateTime>

Specifies a date time value, in D.HH:MM:SS format, for a custom schedule.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Id<String>

Specifies an identifier of an operating system image update schedule in Configuration Manager.

Aliases	Packageld
Required?	true

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies an input object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies the name of an operating system image update schedule in Configuration Manager.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RunNow

Indicates that the schedule is set by ID, input object, or name, and run now.

Aliases	none
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SoftwareUpdate<IResultObject[]>

Specifies an array of input objects.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UpdateDistributionPointsWithImage<Boolean>

Indicates whether to update distribution points with the image.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Utc<Boolean>

Indicates whether to use Coordinated Universal Time (UTC).

Aliases	none
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Set an update schedule

The first command uses the **Get-CMSoftwareUpdate** cmdlet to get the software update associated with the identifier 308, and stores the result in the `$SoftwareUpdate` variable.

The second command uses the **Set-CMOperatingSystemImageUpdateSchedule** cmdlet to set the schedule the update to run now.

```
PS C:\> $SoftwareUpdate = Get-CMSoftwareUpdate -Id "308"
PS C:\> Set-CMOperatingSystemImageUpdateSchedule -SoftwareUpdate $SoftwareUpdate -Name
"Microsoft Windows 8 (x86)" -RunNow -ContinueOnError $True -
UpdateDistributionPointsWithImage $True
```

Example 2: Set a custom update schedule

This example sets a custom update schedule by using the *CustomSchedule* parameter to specify a date and time to run the update.

The first command uses the **Get-CMSoftwareUpdate** cmdlet to get the software updated associated with the identifier 308, and stores the value in the `$SoftwareUpdate` variable.

The second command sets the update schedule and specifies a date time to run the update. The command also specifies UTC time.

```
PS C:\> $SoftwareUpdate = Get-CMSoftwareUpdate -Id "308"
PS C:\> Set-CMOperatingSystemImageUpdateSchedule -SoftwareUpdate $SoftwareUpdate -Name
"Microsoft Windows 8 (x86)" -CustomSchedule "2012/11/17 10:00:00" -Utc $True
```

Related topics

[Clear-CMOperatingSystemImageUpdateSchedule](#)

[Get-CMOperatingSystemImageUpdateSchedule](#)

Set-CMOperatingSystemInstaller

Set-CMOperatingSystemInstaller

Changes configuration settings of operating system installers.

Syntax

Parameter Set: SetById

```
Set-CMOperatingSystemInstaller -Id <String[]> [-Description <String> ] [-NewName <String> ]  
[-Path <String> ] [-SecuredScopeNames <String> ] [-Version <String> ] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SetByName

```
Set-CMOperatingSystemInstaller -Name <String> [-Description <String> ] [-NewName <String> ]  
[-Path <String> ] [-SecuredScopeNames <String> ] [-Version <String> ] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SetByValueMandatory

```
Set-CMOperatingSystemInstaller -InputObject <IResultObject> [-Description <String> ] [-  
NewName <String> ] [-Path <String> ] [-Version <String> ] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SetSecurityScopeById

```
Set-CMOperatingSystemInstaller -Id <String[]> -SecurityScopeAction {AddMembership |  
RemoveMembership} -SecurityScopeName <String> [-SecuredScopeNames <String> ] [-Confirm] [-  
WhatIf] [ <CommonParameters>]
```

Parameter Set: SetSecurityScopeByName

```
Set-CMOperatingSystemInstaller -Name <String> -SecurityScopeAction {AddMembership |  
RemoveMembership} -SecurityScopeName <String> [-SecuredScopeNames <String> ] [-Confirm] [-  
WhatIf] [ <CommonParameters>]
```

Parameter Set: SetSecurityScopeByValue

```
Set-CMOperatingSystemInstaller -InputObject <IResultObject> -SecurityScopeAction  
{AddMembership | RemoveMembership} -SecurityScopeName <String> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Detailed Description

The **Set-CMOperatingSystemInstaller** cmdlet changes configuration settings of one or more operating system installers in Microsoft System Center 2012 Configuration Manager. An operating system installer is an installation package that contains all the files that System Center 2012 Configuration Manager needs to install a Windows operating system on a reference computer.

Parameters

-Description<String>

Specifies a description for the operating system installer.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Id<String[]>

Specifies an array of IDs of operating system installers.

Aliases	Packageld
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a **CMOperatingSystemInstaller** object. To obtain a **CMOperatingSystemInstaller** object, use the **Get-CMOperatingSystemInstaller** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-Name<String>

Specifies the name of an operating system installer.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-NewName<String>

Specifies the new name of an operating system installer.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Path<String>

Specifies the network path to the installation source files of an operating system installer.

Aliases	PackageSourcePath
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-SecuredScopeNames<String>

Specifies names of security scopes. A security scope name can be Default or the name of a custom-created security scope.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SecurityScopeAction<SecurityScopeActionType>

Specifies an action for a security scope. Valid values are: AddMembership and RemoveMembership. The acceptable values for this parameter are:

AddMembership	
RemoveMembership	

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SecurityScopeName<String>

Specifies the name of a security scope. A security scope name can be Default or the name of a custom security scope.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Version<String>

Specifies the version of an operating system installer.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Change settings for an operating system installer by using a name

This command changes configuration settings of the operating system installer named Win8x64. The command renames the operating system installer, adds a version and description, and specifies the path to the installation source files of the operating system installer.

```
PS C:\> Set-CMOperatingSystemInstaller -Name "Win8x64" -NewName "OsiWin8x64" -Version "I20B" -Description "Dept02 Sys Install" -Path "\\Win2k3X64contoso\Public\OSD\win8x64"
```

Example 2: Add an operating system installer to a security scope by using a name

This command adds membership to the security scope named SecScope02 for the operating system installer named InstPkg01.

```
PS C:\> Set-CMOperatingSystemInstaller -SecurityScopeAction AddMembership -SecurityScopeName "SecScope02" -Name "InstPkg01"
```

Example 3: Remove an operating system installer from a security scope

This command removes membership to the security scope named SecScope02 for the operating system installer named InstPkg01.

```
PS C:\> Set-CMOperatingSystemInstaller -SecurityScopeAction RemoveMembership -SecurityScopeName "SecScope02" -Name "InstPkg01"
```

Related topics

[Get-CMOperatingSystemInstaller](#)

[New-CMOperatingSystemInstaller](#)

[Remove-CMOperatingSystemInstaller](#)

Set-CMOutOfBandManagementComponent

Set-CMOutOfBandManagementComponent

Sets the site system server that hosts the out of band management role in System Center 2012 Configuration Manager.

Syntax

Parameter Set: SearchBySiteCodeMandatory

```
Set-CMOutOfBandManagementComponent -SiteCode <String> [-AddAmtUserAccount <String[]> ] [-AllowPingResponse <Boolean> ] [-AmtAccountOU <String> ] [-AmtProvisioningAccounts <Dictionary[]<String><Object>> ] [-AmtProvisioningRemovalAccount <String> ] [-AmtProvisioningRemovalPassword <SecureString> ] [-AmtProvisioningSchedule <IResultObject> ] [-AuditLogSettingName {AgentPresenceManager | CircuitBreakerManager | EndpointAccessControl | EventManager | FirmwareUpdateManager | NetworkAdministration | NetworkTime | RedirectionManager | RemoteControlOperations | SecurityAdministration | SecurityAuditLog | StorageAdministration | WirelessConfiguration} ] [-CertificateTemplate <String> ] [-CertificationAuthorityName <String> ] [-EnableBypassBiosPassword <Boolean> ] [-EnableCrlChecking <Boolean> ] [-EnableIDERedirection <Boolean> ] [-EnableWebInterface <Boolean> ] [-EnableWiredNetworkAccess <Boolean> ] [-EnrollmentPoint <String> ] [-IssuingCertificationAuthority <String> ] [-KerberosClockToleranceMinutes <Int32> ] [-MebxAccount <String> ] [-MebxPassword <SecureString> ] [-PowerState {AlwaysOnS0S5 | HostIsOnS0} ] [-RemoveAmtUserAccount <String[]> ] [-SiteSystemServerName <String[]> ] [-UniversalSecurityGroup <String> ] [-WiredProfileObject <WiredProfile> ] [-WirelessProfile <WirelessProfile[]> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Set-CMOutOfBandManagementComponent** cmdlet sets the site system computer that hosts the out of band management role in Microsoft System Center 2012 Configuration Manager. The out of band management role manages computers that have the Intel vPro chip set and a version of Intel Active Management Technology (AMT) that System Center 2012 Configuration Manager supports. Out of band management lets you connect to a computer AMT management controller when the computer is turned off, in hibernation, or otherwise unresponsive through the operating system.

Parameters

-AddAmtUserAccount<String[]>

Specifies an array of AMT user accounts to add.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AllowPingResponse<Boolean>

Indicates whether to allow ping responses.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AmtAccountOU<String>

Specifies an organizational unit (OU) for an AMT account.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AmtProvisioningAccounts<Dictionary[]<String><Object>>

Specifies an array of key-value pairs.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AmtProvisioningRemovalAccount<String>

Specifies an AMT account.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AmtProvisioningRemovalPassword<SecureString>

Specifies a secure string that contains a password.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AmtProvisioningSchedule<IResultObject>

Specifies an input object. To obtain an input object, use the **New-CMSchedule** cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AuditLogSettingName<AuditLogSettingType[]>

Specifies an array of audit log setting names. Valid values are:

- AgentPresenceManager
- CircuitBreakerManager
- EndpointAccessControl
- EventManager
- FirmwareUpdateManager
- NetworkAdministration
- NetworkTime
- RedirectionManager
- RemoteControlOperations
- SecurityAdministration
- SecurityAuditLog
- StorageAdministration
- WirelessConfiguration

The acceptable values for this parameter are:

AgentPresenceManager	
CircuitBreakerManager	
EndpointAccessControl	
EventManager	
FirmwareUpdateManager	
NetworkAdministration	
NetworkTime	
RedirectionManager	
RemoteControlOperations	

SecurityAdministration	
SecurityAuditLog	
StorageAdministration	
WirelessConfiguration	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CertificateTemplate<String>

Specifies a certificate template.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CertificationAuthorityName<String>

Specifies the name of a certification authority.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnableBypassBiosPassword<Boolean>

Indicates whether to bypass the BIOS password.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnableCrlChecking<Boolean>

Indicates whether to enable certificate revocation list (CRL) checking.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnableIDERedirection<Boolean>

Indicates whether to enable IDE redirection. Intel AMT uses IDE redirection to redirect serial and IDE communication from a managed client to a management console.

Aliases	none
Required?	false
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnableWebInterface<Boolean>

Indicates whether to enable the web interface.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnableWiredNetworkAccess<Boolean>

Indicates whether to enable wired network access.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnrollmentPoint<String>

Specifies an enrollment point in Configuration Manager.

Aliases	none
Required?	false
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-IssuingCertificationAuthority<String>

Specifies the issuing certification authority.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-KerberosClockToleranceMinutes<Int32>

Specifies a clock tolerance, in minutes, for Kerberos. Kerberos authentication depends on time synchronization.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-MebxAccount<String>

Specifies the name of an account for Management Engine BIOS Extensions (MEBx). The MEBx account provides authenticated access to the AMT firmware on AMT-based computers.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-MebxPassword<SecureString>

Specifies a secure string that contains the password for the MEBx account.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PowerState<PowerStateType>

Specifies an AMT power state that describes system states as on, sleeping, hibernating, or off. Valid values are:

-- AlwaysOnS0S5

-- HostIsOnS0

The acceptable values for this parameter are:

AlwaysOnS0S5	
HostIsOnS0	

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RemoveAmtUserAccount<String[]>

Specifies an array of AMT user accounts to remove.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteCode<String>

Specifies a site code in Configuration Manager.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteSystemServerName<String[]>

Specifies an array of names of site system servers in Configuration Manager.

Aliases	Name
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UniversalSecurityGroup<String>

Specifies the name of a universal security group.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-WiredProfileObject<WiredProfile>

Specifies a wired profile object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-WirelessProfile<WirelessProfile[]>

Specifies an array of wireless profiles.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Set an out of band management component

The first command uses the **Get-CMTrustedRootCertificate** cmdlet to get a certificate, and stores the certificate in the \$Cert variable.

The second command uses the **New-CMWiredProfileObject** cmdlet to create a profile object, and stores the object in the \$WiredP variable.

The third command uses the **New-CMWirelessProfileObject** cmdlet to create a wireless profile object, and stores the object in the \$WirelessP variable.

The fourth command uses the **Set-CMOutOfBandManagementComponent** cmdlet to set an out of band management component by using the \$WiredP and \$WirelessP variables.

```
PS C:\> $Cert = Get-CMTrustedRootCertificate -CertificationAuthorityServerName
"CertAuth.Contoso.Com"
PS C:\> $WiredP = New-CMWiredProfileObject -TrustedRootCertificate $Cert -
ClientAuthenticationMethod EapTtlsMschapv2 -ClientIssuingCertificationAuthority
"ContosoCorpTPM.Contoso.Com" -ClientCertificationAuthorityName "Contoso TPM" -
ClientCertificateTemplate "Contoso Certificate Access" - MachineAuth - TPM"
PS C:\> $WirelessP = New-CMWirelessProfileObject -ProfileName "Test -NetworkName Net1" -
SecurityType WPA2Enterprise -EncryptionMethod AES -TrustedRootCertificate $Cert -
ClientAuthenticationMethod EapTtlsMschapv2 -ClientIssuingCertificationAuthority
"ContosoCorpTPM.Contoso.Com" -ClientCertificationAuthorityName "Contoso TPM" -
ClientCertificateTemplate "Contoso Certificate Access" - MachineAuth - TPMv2"
PS C:\> Set-CMOutOfBandManagementComponent -SiteCode "CM2" -EnableWiredNetworkAccess $True
-WiredProfileObject $WiredP -WirelessProfile $WirelessP
```

Example 2: Set an out of band management component with an AMT provisioning account

The first command creates a password string for the AMT provisioning account. The command uses a secure string to obscure the password.

The second command uses the **New-CMAmtProvisioningAccount** cmdlet to create an account, and stores the result in the \$Apa variable.

The third command uses the **New-CMSchedule** cmdlet to create a schedule, and stores the result in the \$Schedule variable.

The fourth command uses the **Set-CMOutOfBandManagementComponent** cmdlet to set an out of band management component, by using the \$Apa and \$Schedule variables. The command specifies an account for the *AmtProvisioningAccount* parameter.

```
PS C:\> $SS= Read-Host -AsSecureString
PS C:\> $Apa = New-CMAmtProvisioningAccount -UserName "ElisaDaugherty" -Password $SS -
Description "New AMT Provisioning Account"
PS C:\> $Schedule = New-CMSchedule -DayOfWeek Saturday -RecurCount 2 -Start "2012/11/22
12:10:00"
PS C:\> Set-CMOutOfBandManagementComponent -SiteCode CM2 -AmtProvisioningAccount $Apa -
AmtProvisioningSchedule $Schedule -AmtProvisioningRemovalAccount "Western\ElisaDaugherty" -
AmtProvisioningRemovalPassword $SS
```

Example 3: Set an out of band management component with an AMT user account

The first command creates a password string for the AMT provisioning account. The command uses a secure string to obscure the password.

The second command uses the **New-CMAmtProvisioningAccount** cmdlet to create an account, and stores the result in the \$Apa variable.

The third command uses the **New-CMSchedule** cmdlet to create a schedule, and stores the result in the \$Schedule variable.

The fourth command uses the **Set-CMOutOfBandManagementComponent** cmdlet to set an out of band management component, by using the \$Apa and \$Schedule variables. The command specifies an account name for the *AmtUserAccount* parameter.

```
PS C:\> $SS = Read-Host -AsSecureString
PS C:\> $Apa = New-CMAmtProvisioningAccount -UserName "ElisaDaugherty " -Password "$SS" -
Description "New AMT Provisioning Account"
PS C:\> $Schedule= New-CMSchedule -DayOfWeek Saturday -RecurCount 2 -Start "2012/11/22
12:10:00"
PS C:\> Set-CMOutOfBandManagementComponent -SiteCode "CM2" -AmtUserAccount
"Western\ElisaDaugherty " -PowerState HostIsOnS0 -EnableWebInterface $True -
EnableIDERedirection $False -AllowPingResponse $True -EnableBypassBiosPassword $False -
KerberosClockToleranceMinutes 3 -AuditLogSettingName
EndpointAccessControl,CircuitBreakerManager,AgentPresenceManager -AmtProvisioningAccount
$Apa -AmtProvisioningSchedule $Schedule -AmtProvisioningRemovalAccount
"Western\ElisaDaugherty" -AmtProvisioningRemovalPassword $SS
```

Example 4: Set an out of band management component with an AMT account OU

The command uses the **Set-CMOutOfBandManagementComponent** cmdlet to set an out of band management component, and specifies an organizational unit with the *AmtAccountOU* parameter.

```
PS C:\> Set-CMOutOfBandManagementComponent -SiteCode "CM2" -AmtAccountOU
"LDAP://OU=Resources,DC=Western,DC=Contoso,DC=Com" -UniversalSecurityGroup "Administrators"
-IssuingCertificationAuthority "Test.Western.Contoso.Com" -CertificationAuthorityName
"Contoso Test" -CertificateTemplate "Test - Secure Web Server 2yr" -EnableCrlChecking $False
```

Related topics

[Get-CMOutOfBandManagementComponent](#)

[New-CMSchedule](#)

[Get-CMTrustedRootCertificate](#)

[New-CMAmtProvisioningAccount](#)

[New-CMWiredProfileObject](#)

Set-CMOutOfBandServicePoint

Set-CMOutOfBandServicePoint

Changes configuration settings for an out-of-band service point.

Syntax

Parameter Set: SetByName

```
Set-CMOutOfBandServicePoint -SiteCode <String> -SiteSystemServerName <String> [-  
EnableCrlChecking <Boolean> ] [-ErrorRetryCount <Int32> ] [-ErrorRetryMinutesDelay <Int32> ]  
[-ProvisioningCertificateThumbprint <String> ] [-TransmissionStartMinutesInterval <Int32> ]  
[-TransmissionThreadCount <Int32> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetByValue

```
Set-CMOutOfBandServicePoint -InputObject <IResultObject> [-EnableCrlChecking <Boolean> ] [-  
ErrorRetryCount <Int32> ] [-ErrorRetryMinutesDelay <Int32> ] [-  
ProvisioningCertificateThumbprint <String> ] [-TransmissionStartMinutesInterval <Int32> ] [-  
TransmissionThreadCount <Int32> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Set-CMOutOfBandServicePoint** cmdlet changes configuration settings for an out-of-band service point. An out-of-band service point is a site system role that provisions and configures Intel Active Management Technology (AMT)-based computers for Microsoft System Center 2012 Configuration Manager.

Parameters

-EnableCrlChecking<Boolean>

Indicates whether the out-of-band service point verifies the certificate revocation list (CRL) for the provisioning certificate.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ErrorRetryCount<Int32>

Specifies the number of retry attempts that Configuration Manager makes after it fails to turn on an AMT-based computer.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ErrorRetryMinutesDelay<Int32>

Specifies the number of minutes that Configuration Manager waits between retry attempts to turn on an AMT-based computer.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a **CMOutOfBandServicePoint** object. To obtain a **CMOutOfBandServicePoint** object, use the **Get-CMOutOfBandServicePoint** cmdlet.

Aliases	none
Required?	true

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ProvisioningCertificateThumbprint<String>

Specifies the thumbprint of the AMT provisioning certificate.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteCode<String>

Specifies the site code of the Configuration Manager site that hosts the site system role.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteSystemServerName<String>

Specifies a fully qualified domain name (FQDN) of the server that hosts the site system role.

Aliases	none
Required?	true

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-TransmissionStartMinutesInterval<Int32>

Specifies the time, in minutes, that Configuration Manager sends commands to turn on an AMT-based computer before a scheduled wake-up activity.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-TransmissionThreadCount<Int32>

Specifies the maximum number of connection threads that the site system role supports.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Change settings of an out-of-band service point

This command changes the settings of the out-of-band service point from the System Center 2012 Configuration Manager site that has the site code named CM2 on the site system named cmcen-dist02.tsqa.contoso.com.

```
PS C:\> Set-CMOutOfBandServicePoint -SiteSystemServerName "cmcen-dist02.tsqa.contoso.com" -
SiteCode "CM2" -ErrorRetryMinutesDelay 1 -TransmissionThreadCount 1 -
TransmissionStartMinutesInterval 1 -EnableCrlChecking 1 -ProvisioningCertificateThumbprint
"916EC36F1068D47DE48A02A788A9DB137CD0B674"
```

Example 2: Change settings of an out-of-band service point by using an object variable

The first command gets the out-of-band service point from the System Center 2012 Configuration Manager site that has the site code named CM2 on the site system named cmcen-dist02.tsqa.contoso.com. The command stores the results in the \$Osp variable.

The second command changes the settings of the out-of-band service point stored in the \$Osp variable.

```
PS C:\> $Osp = Get-CMOutOfBandServicePoint -SiteSystemServerName "cmcen-  
dist02.tsqa.contoso.com" -SiteCode "CM2"  
PS C:\> Set-CMOutOfBandServicePoint -InputObject $Osp -ErrorRetryCount 1 -  
ErrorRetryMinutesDelay 1 -TransmissionThreadCount 1 -TransmissionStartMinutesInterval 1 -  
EnableCrlChecking 1 -ProvisioningCertificateThumbprint  
"916EC36F1068D47DE48A02A788A9DB137CD0B674"
```

Related topics

[Get-CMOutOfBandServicePoint](#)

[Add-CMOutOfBandServicePoint](#)

[Remove-CMOutOfBandServicePoint](#)

Set-CMPackage

Set-CMPackage

Changes configuration settings for Configuration Manager packages.

Syntax

Parameter Set: SetById

```
Set-CMPackage -Id <String[]> [-Description <String> ] [-DistributionPriority {High | Low | Normal} ] [-ForcedDisconnectDelay <Int32> ] [-ForcedDisconnectEnabled <Boolean> ] [-ForcedDisconnectNumberRetries <Int32> ] [-Language <String> ] [-Manufacturer <String> ] [-MifFileName <String> ] [-MifName <String> ] [-MifPublisher <String> ] [-MifVersion <String> ] [-NewName <String> ] [-Path <String> ] [-SecuredScopeNames <String> ] [-ShareName <String> ] [-ShareType {ShareCommon | ShareSpecific} ] [-Version <String> ] [-Confirm] [-WhatIf] [ <CommonParameters> ]
```

Parameter Set: SetByName

```
Set-CMPackage -Name <String> [-Description <String> ] [-DistributionPriority {High | Low | Normal} ] [-ForcedDisconnectDelay <Int32> ] [-ForcedDisconnectEnabled <Boolean> ] [-ForcedDisconnectNumberRetries <Int32> ] [-Language <String> ] [-Manufacturer <String> ] [-MifFileName <String> ] [-MifName <String> ] [-MifPublisher <String> ] [-MifVersion <String> ] [-NewName <String> ] [-Path <String> ] [-SecuredScopeNames <String> ] [-ShareName <String> ] [-ShareType {ShareCommon | ShareSpecific} ] [-Version <String> ] [-Confirm] [-WhatIf] [ <CommonParameters> ]
```

Parameter Set: SetByValue

```
Set-CMPackage -InputObject <IResultObject> [-Description <String> ] [-DistributionPriority {High | Low | Normal} ] [-ForcedDisconnectDelay <Int32> ] [-ForcedDisconnectEnabled <Boolean> ] [-ForcedDisconnectNumberRetries <Int32> ] [-Language <String> ] [-Manufacturer <String> ] [-MifFileName <String> ] [-MifName <String> ] [-MifPublisher <String> ] [-MifVersion <String> ] [-NewName <String> ] [-Path <String> ] [-ShareName <String> ] [-ShareType {ShareCommon | ShareSpecific} ] [-Version <String> ] [-Confirm] [-WhatIf] [ <CommonParameters> ]
```

Detailed Description

The **Set-CMPackage** cmdlet changes configuration settings for Microsoft System Center 2012 Configuration Manager packages.

If you set the *MifFileName* parameter, System Center 2012 Configuration Manager looks in the %TEMP% directory or the %windir% directory for the installation status Management Information Format (MIF) file that you specified in *MifFileName*. The installation status indicates whether the program successfully ran.

If System Center 2012 Configuration Manager does not find the file, it searches for all MIF files in those directories. System Center 2012 Configuration Manager makes a case-insensitive comparison of the values that you specify for *MifName*, *MifPublisher*, and *MifVersion* to the values that the MIF file specifies. If System Center 2012 Configuration Manager finds a match, it uses the status that the MIF file specifies as the installation status for the program. If System Center 2012 Configuration Manager cannot find a match, or if you do not specify *MifFileName*, System Center 2012 Configuration Manager uses the program exit code to set the installation status for the program. An exit code of zero indicates that the program successfully ran. Any other values indicate application-specific error codes.

Parameters

-Description<String>

Specifies a description for the package. You can use a maximum of 128 characters.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DistributionPriority<Priorities>

Specifies the sending priority for the deployment package. Configuration Manager uses the sending priority for the deployment package when it sends the package to distribution points at child sites. Valid values are:

- High
- Medium
- Low

Configuration Manager sends packages in priority order. Configuration Manager sends packages that have the same priority in order of creation. Unless a backlog exists, Configuration Manager sends a package immediately regardless of its priority.

The acceptable values for this parameter are:

High	
Low	
Normal	

Aliases	Priority
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ForcedDisconnectDelay<Int32>

Specifies the time, in minutes, that Configuration Manager waits before it forcibly disconnects users from the distribution point share. The default value is 5.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ForcedDisconnectEnabled<Boolean>

Indicates whether Configuration Manager forcibly disconnects users from the distribution point share when a share violation occurs in updating, refreshing, or deleting package source files.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ForcedDisconnectNumberRetries<Int32>

Specifies the number of times Configuration Manager attempts to disconnect a user from the distribution point share. The default value is 2.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Id<String[]>

Specifies an array of package IDs.

Aliases	PackageId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a **CMPackage** object. To obtain a **CMPackage** object, use the [Get-CMPackage](#) cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Language<String>

Specifies the language version of the package. You can use a maximum of 32 characters in a format that you choose to use to identify the language version. Configuration Manager uses the Language property together with Manufacturer, Name, and Version to identify a package. For example, you can have an English version and a German version of the same package.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Manufacturer<String>

Specifies a manufacturer name to help you identify the package. You can use a maximum of 32 characters.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-MifFileName<String>

Specifies the name of the MIF file that contains the package status.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-MifName<String>

Specifies the name of the MIF file that contains the program status for the package. The file name extension must be .mif.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-MifPublisher<String>

Specifies the name of the software publisher of the package.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-MifVersion<String>

Specifies the version number of the MIF file.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies a package name.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-NewName<String>

Specifies a new name for the package.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Path<String>

Specifies the location of the files of update contents for the package.

You can specify either a full local path or a Universal Naming Convention (UNC) path. Make sure that this location contains all the files and subdirectories that the program needs to run, including any scripts.

Aliases	PackageSourcePath
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SecuredScopeNames<String>

Specifies names of security scopes. A security scope name can be Default or the name of a custom-created security scope.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ShareName<String>

Specifies the UNC share location to use on the distribution point. You can include directories in the share name. If the directories do not exist, Configuration Manager creates them. You must specify a share name if you set the *ShareType* parameter to SHARE_SPECIFIC.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ShareType<ShareTypes>

Specifies the type of share that the distribution point uses when it shares the package. Valid values are:

-- SHARE_COMMON

-- SHARE_SPECIFIC

The default value is SHARE_COMMON. If you specify SHARE_SPECIFIC, you must specify a value for the *ShareName* parameter.

The acceptable values for this parameter are:

ShareCommon	
ShareSpecific	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Version<String>

Specifies a version number for the package.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Rename a package and add a description

This command renames the package that has the ID ST120001. The command changes the name of the package to ScriptsPackage02 and adds a description for the package.

```
PS C:\> Set-CMPackage -Id "ST120001" -NewName "ScriptsPackage02" -Description "This package
deploys scripts that run on a recurring schedule."
```

Example 2: Rename a package by using an object variable

The first command gets the package that has the ID ST120001, and stores the results in the \$Pkg variable.

The second command changes the name of the package stored in \$Pkg to ScriptsPackage02, and adds a description for the package.

```
PS C:\> $Pkg = Get-CMPackage -Id ST120001
PS C:\> Set-CMPackage -InputObject $Pkg -Newname "ScriptsPackage02" -Description "This
package deploys scripts that run on a recurring schedule."
```

Related topics

[Export-CMPackage](#)

[Get-CMPackage](#)

[Import-CMPackage](#)

[New-CMPackage](#)

[Remove-CMPackage](#)

Set-CMPackageDeployment

Set-CMPackageDeployment

Changes values that define how Configuration Manager deploys a software package.

Syntax

Parameter Set: SetDeviceProgramDeploymentByPackageId

```
Set-CMPackageDeployment -CollectionName <String> -DeviceProgramName <String> -PackageId <String> [-Comment <String> ] [-DeploymentStartDay <DateTime> ] [-DeploymentStartTime <DateTime> ] [-RecurUnit {Days | Hours | Minutes} ] [-RecurValue <Int32> ] [-Rerun <Boolean> ] [-UseMeteredNetwork <Boolean> ] [-UseRecurrencePattern <Boolean> ] [-UseUtc <Boolean> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetDeviceProgramDeploymentByPackageName

```
Set-CMPackageDeployment -CollectionName <String> -DeviceProgramName <String> -PackageName <String> [-Comment <String> ] [-DeploymentStartDay <DateTime> ] [-DeploymentStartTime <DateTime> ] [-RecurUnit {Days | Hours | Minutes} ] [-RecurValue <Int32> ] [-Rerun <Boolean> ] [-UseMeteredNetwork <Boolean> ] [-UseRecurrencePattern <Boolean> ] [-UseUtc <Boolean> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetDeviceProgramDeploymentByPackageValue

```
Set-CMPackageDeployment -CollectionName <String> -Comment <String> -DeviceProgramName <String> -Package <IResultObject> [-DeploymentStartDay <DateTime> ] [-DeploymentStartTime <DateTime> ] [-RecurUnit {Days | Hours | Minutes} ] [-RecurValue <Int32> ] [-Rerun <Boolean> ] [-UseMeteredNetwork <Boolean> ] [-UseRecurrencePattern <Boolean> ] [-UseUtc <Boolean> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetStandardProgramDeploymentByPackageId

```
Set-CMPackageDeployment -CollectionName <String> -PackageId <String> -StandardProgramName <String> [-AllowSharedContent <Boolean> ] [-AllowUsersRunIndependently <Boolean> ] [-Comment <String> ] [-DeploymentAvailableDay <DateTime> ] [-DeploymentAvailableTime <DateTime> ] [-DeploymentExpireDay <DateTime> ] [-DeploymentExpireTime <DateTime> ] [-EnableExpireSchedule <Boolean> ] [-FastNetworkOption {DownloadContentFromDistributionPointAndRunLocally | RunProgramFromDistributionPoint} ] [-PersistOnWriteFilterDevice <Boolean> ] [-RerunBehavior {AlwaysRerunProgram | NeverRerunDeployedProgram | RerunIfFailedPreviousAttempt | RerunIfSucceededOnPreviousAttempt} ] [-Schedule <IResultObject[]> ] [-ScheduleEvent {AsSoonAsPossible | LogOff | LogOn} ] [-SendWakeUpPacket <Boolean> ] [-SlowNetworkOption {DoNotRunProgram | DownloadContentFromDistributionPointAndLocally | RunProgramFromDistributionPoint} ] [-SoftwareInstallation <Boolean> ] [-SystemRestart <Boolean> ] [-UseMeteredNetwork <Boolean> ] [-UseUtcForAvailableSchedule <Boolean> ] [-UseUtcForExpireSchedule <Boolean> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetStandardProgramDeploymentByPackageName

```
Set-CMPackageDeployment -CollectionName <String> -PackageName <String> -StandardProgramName <String> [-AllowSharedContent <Boolean> ] [-AllowUsersRunIndependently <Boolean> ] [-Comment <String> ] [-DeploymentAvailableDay <DateTime> ] [-DeploymentAvailableTime <DateTime> ] [-
```

```

DeploymentExpireDay <DateTime> ] [-DeploymentExpireTime <DateTime> ] [-EnableExpireSchedule
<Boolean> ] [-FastNetworkOption {DownloadContentFromDistributionPointAndRunLocally |
RunProgramFromDistributionPoint} ] [-PersistOnWriteFilterDevice <Boolean> ] [-RerunBehavior
{AlwaysRerunProgram | NeverRerunDeployedProgram | RerunIfFailedPreviousAttempt |
RerunIfSucceededOnPreviousAttempt} ] [-Schedule <IResultObject[]> ] [-ScheduleEvent
{AsSoonAsPossible | LogOff | LogOn} ] [-SendWakeUpPacket <Boolean> ] [-SlowNetworkOption
{DoNotRunProgram | DownloadContentFromDistributionPointAndLocally |
RunProgramFromDistributionPoint} ] [-SoftwareInstallation <Boolean> ] [-SystemRestart
<Boolean> ] [-UseMeteredNetwork <Boolean> ] [-UseUtcForAvailableSchedule <Boolean> ] [-
UseUtcForExpireSchedule <Boolean> ] [-Confirm] [-WhatIf] [ <CommonParameters>]

```

```

Parameter Set: SetStandardProgramDeploymentByPackageValue
Set-CMPackageDeployment -CollectionName <String> -Package <IResultObject> -
StandardProgramName <String> [-AllowSharedContent <Boolean> ] [-AllowUsersRunIndependently
<Boolean> ] [-Comment <String> ] [-DeploymentAvailableDay <DateTime> ] [-
DeploymentAvailableTime <DateTime> ] [-DeploymentExpireDay <DateTime> ] [-
DeploymentExpireTime <DateTime> ] [-EnableExpireSchedule <Boolean> ] [-FastNetworkOption
{DownloadContentFromDistributionPointAndRunLocally | RunProgramFromDistributionPoint} ] [-
PersistOnWriteFilterDevice <Boolean> ] [-RerunBehavior {AlwaysRerunProgram |
NeverRerunDeployedProgram | RerunIfFailedPreviousAttempt |
RerunIfSucceededOnPreviousAttempt} ] [-Schedule <IResultObject[]> ] [-ScheduleEvent
{AsSoonAsPossible | LogOff | LogOn} ] [-SendWakeUpPacket <Boolean> ] [-SlowNetworkOption
{DoNotRunProgram | DownloadContentFromDistributionPointAndLocally |
RunProgramFromDistributionPoint} ] [-SoftwareInstallation <Boolean> ] [-SystemRestart
<Boolean> ] [-UseMeteredNetwork <Boolean> ] [-UseUtcForAvailableSchedule <Boolean> ] [-
UseUtcForExpireSchedule <Boolean> ] [-Confirm] [-WhatIf] [ <CommonParameters>]

```

Detailed Description

The **Set-CMPackageDeployment** cmdlet changes values that define how Microsoft System Center 2012 Configuration Manager deploys a software package. A deployment includes a collection of devices or users, a package to deploy, and either a device program name or a standard program name. To specify which deployment to modify, specify the collection name, package, and program name. You can specify the package by name or ID, or you can use the **Get-CMPackage** cmdlet to get a package object.

Parameters

-AllowSharedContent<Boolean>

Indicates whether clients use shared content. If this value is \$True, clients attempt to download content from other clients that downloaded that content. If this value is \$False, clients do not attempt to download from other clients.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AllowUsersRunIndependently<Boolean>

Indicates whether users can install the software independently. If this value is \$True, users can install software in this package from the software library regardless of the scheduled installation time. If this value is \$False, the software installs at the scheduled time.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CollectionName<String>

Specifies the ID of a device or user collection.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Comment<String>

Specifies a comment for the deployment.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeploymentAvailableDay<DateTime>

Specifies a day as a **DateTime** object. To obtain a **DateTime** object, use the **Get-Date** cmdlet. For more information, type `Get-Help Get-Date`. This is the day on which the deployment becomes available. If you specify a value for the *DeployAvailableTime* parameter in addition to this parameter, the cmdlet uses that value.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeploymentAvailableTime<DateTime>

Specifies a time as a **DateTime** object. To obtain a **DateTime** object, use the **Get-Date** cmdlet. This is the time at which the deployment becomes available.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeploymentExpireDay<DateTime>

Specifies a day as a **DateTime** object. To obtain a **DateTime** object, use the **Get-Date** cmdlet. This is the day on which the deployment expires. If you specify a value for the *DeploymentExpireTime* parameter in addition to this parameter, the cmdlet uses that value.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeploymentExpireTime<DateTime>

Specifies a time as a **DateTime** object. To obtain a **DateTime** object, use the **Get-Date** cmdlet. This is the time at which the deployment expires.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeploymentStartDay<DateTime>

Specifies a day as a **DateTime** object. To obtain a **DateTime** object, use the **Get-Date** cmdlet. This is the day on which the deployment starts. If you specify a value for the *DeploymentStartTime* parameter in addition to this parameter, the cmdlet uses that value.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeploymentStartTime<DateTime>

Specifies a time as a **DateTime** object. To obtain a **DateTime** object, use the **Get-Date** cmdlet. This is the time at which the deployment starts.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeviceProgramName<String>

Specifies the name of a device program.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnableExpireSchedule<Boolean>

Indicates whether to enable the schedule to expire the deployment.

Aliases	none
Required?	false
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-FastNetworkOption<FastNetworkOptionType>

Specifies client behavior on a fast network. Valid values are:

- DownloadContentFromDistributionPointAndRunLocally
- RunProgramFromDistributionPoint

The acceptable values for this parameter are:

DownloadContentFromDistributionPointAndRunLocally	
RunProgramFromDistributionPoint	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Package<IResultObject>

Specifies a package object. To obtain a package object, use the **Get-CMPackage** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PackageId<String>

Specifies the ID of a package.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PackageName<String>

Specifies the name of a package.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PersistOnWriteFilterDevice<Boolean>

Indicates whether to enable write filters for embedded devices. For a value of \$True, the device commits changes during a maintenance window. This action requires a restart. For a value of \$False, the device saves changes in an overlay and commits them later.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RecurUnit<RecurUnitType>

Specifies a unit for a recurring deployment. Valid values are:

- Days
- Hours
- Minutes

The acceptable values for this parameter are:

Days	
Hours	
Minutes	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RecurValue<Int32>

Specifies how often a deployment recurs. This parameter depends on the unit type specified in the *RecurUnit* parameter. This value can be between 1 and 23 if the unit is Hours, between 1 and 31 if the unit is Days, or between 1 and 59 if the unit is Minutes.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Rerun<Boolean>

Indicates whether the deployment reruns. If this value is \$True, the deployment runs again for clients as specified in the *RerunBehavior* parameter. If this value is \$False, the deployment does not run again.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RerunBehavior<RerunBehaviorType>

Specifies how a deployment reruns on a client. Valid values are:

-- AlwaysRetunProgram. Rerun as scheduled, even if the deployment succeeded. You can use this value for recurring deployments.

-- NeverRerunDeployedProgram. Does not rerun, even if the deployment failed or files changed.

-- RerunIfFailedPreviousAttempt. Rerun, as scheduled, if the deployment failed on the previous attempt.

-- RerunIfSucceededOnpreviousAttempt. Rerun only if the previous attempt succeeded. You can use this value for updates that depend on the previous update.

The acceptable values for this parameter are:

AlwaysRerunProgram	
NeverRerunDeployedProgram	
RerunIfFailedPreviousAttempt	
RerunIfSucceededOnPreviousAttempt	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Schedule<IResultObject[]>

Specifies an array of schedule objects for the deployment.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ScheduleEvent<ScheduleEventType[]>

Specifies an array of schedule event types. Valid values are:

- AsSoonAsPossible
- LogOff
- LogOn
- SendWakeUpPacket

The acceptable values for this parameter are:

AsSoonAsPossible	
LogOff	
LogOn	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SendWakeUpPacket<Boolean>

Indicates whether to send a wake up packet to computers before the deployment begins. If this value is \$True, Configuration Manager wakes a computer from sleep. If this value is \$False, it does not wake computers from sleep. For computers to wake, you must first configure Wake On LAN.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SlowNetworkOption<SlowNetworkOptionType>

Specifies how Configuration Manager deploys this package in a slow network. Valid values are:

- DoNotRunProgram
- DownloadContentFromDistributionPointAndLocally
- RunProgramFromDistributionPoint

The acceptable values for this parameter are:

DoNotRunProgram	
DownloadContentFromDistributionPointAndLocally	
RunProgramFromDistributionPoint	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SoftwareInstallation<Boolean>

Indicates whether to install the deployed software outside of maintenance windows. A maintenance window is a specified period of time used for computer maintenance and updates. If this value is \$True, the Configuration Manager installs software according to schedule, even if the schedule falls outside a maintenance window. If this value is \$False, Configuration Manager does not install deployed software outside any windows, but waits for a maintenance window.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-StandardProgramName<String>

Specifies a standard program name.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SystemRestart<Boolean>

Indicates whether a system restarts outside a maintenance window. A maintenance window is a specified period of time used for computer maintenance and updates. If this value is \$True, any required restart takes place without regard to maintenance windows. If this value is \$False, the computer does not restart outside a maintenance window.

Aliases	none
Required?	false
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UseMeteredNetwork<Boolean>

Indicates whether to allow clients to download content over a metered Internet connection after the deadline, which may incur additional expense.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UseRecurrencePattern<Boolean>

Indicates whether to use a recurrence pattern.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UseUtc<Boolean>

Indicates whether to use Coordinated Universal Time (UTC), also known as Greenwich Mean Time. If this value is \$True, Configuration Manager uses UTC for this deployment. If this value is \$False, Configuration Manager uses local time.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UseUtcForAvailableSchedule<Boolean>

Indicates whether to use UTC for available schedule. If this value is \$True, Configuration Manager uses UTC. If this value is \$False, Configuration Manager uses local time.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UseUtcForExpireSchedule<Boolean>

Indicates whether to use UTC for expire schedule. If this value is \$True, Configuration Manager uses UTC. If this value is \$False, Configuration Manager uses local time.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Set recurrence properties

This command makes changes to the deployment specified by the collection named All Systems, the device program named DPM, and the package named User State Migration Tool for Windows 8. The command sets the *UseRecurrencePattern* parameter to a value of \$True. The command specifies a recur unit of Hours and a recur value of seven, therefore the deployment recurs every seven hours.

```
PS C:\> Set-CMPackageDeployment -CollectionName "All Systems" -DeviceProgramName "DPM" -  
PackageName "User State Migration Tool for Windows 8" -RecurUnit Hours -RecurValue 7 -  
UseRecurrencePattern $True
```

Example 2: Set availability day and time

This command makes changes to the deployment specified by the collection named All Systems, the package named User State Migration Tool for Windows 8, and the standard program named SPM. The

command specifies a day and time when the deployment becomes available. The command also specifies that the deployment does not use UTC for the availability schedule. The schedule refers to the local time zone.

```
PS C:\> Set-CMPackageDeployment -CollectionName "All Systems" -PackageName "User State Migration Tool for Windows 8" -StandardProgramName "SPM" -DeploymentAvailableDay 2012/10/18 -DeploymentAvailableTime 15:41 -UseUtcForAvailableSchedule $False
```

Related topics

[Start-CMPackageDeployment](#)

[Get-CMPackage](#)

Set-CMPowerControl

Set-CMPowerControl

Changes power state for client devices by using AMT power control commands.

Syntax

Parameter Set: SearchByNameMandatory

```
Set-CMPowerControl -DeviceName <String[]> -PowerControl {None | Restart | Shutdown | Wakeup} [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Set-CMPowerControl -DeviceId <String[]> -PowerControl {None | Restart | Shutdown | Wakeup} [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Set-CMPowerControl -InputObject <IResultObject> -PowerControl {None | Restart | Shutdown | Wakeup} [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Set-CMPowerControl** cmdlet changes the power state for one or more Intel Active Management Technology (Intel AMT) provisioned client devices in Microsoft System Center 2012 Configuration Manager by using AMT power control commands.

Parameters

-DeviceId<String[]>

Specifies an array of device IDs.

Aliases	ResourceId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeviceName<String[]>

Specifies an array of device names.

Aliases	Name
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a **CMPowerControl** object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PowerControl<PowerControlType>

Specifies the power management action. Valid values are:

- None. Disables power settings.
- WakeUp. Turns on a sleeping computer.
- Restart. Performs a hard reset of the computer and turns on the computer. This action does not shut the operating system down.
- Shutdown. Performs a hard reset of the computer. This action does not shut the operating system down.

The acceptable values for this parameter are:

None	
------	--

Restart	
Shutdown	
Wakeup	

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Change the power control setting for a client device

This command changes the power control setting to Restart for the client device that has the ID 209224563.

```
PS C:\> Set-CMPowerControl -DeviceId "209224563" -PowerControl Restart
```

Set-CMProgram

Set-CMProgram

Modifies a program in Configuration Manager.

Syntax

Parameter Set: SetStandardProgramByName

```
Set-CMProgram -Name <String> -StandardProgramName <String> [-AfterRunningType  
{ConfigurationManagerLogsUserOff | ConfigurationManagerRestartsComputer | NoActionRequired |  
ProgramControlsRestart} ] [-Category <String> ] [-CommandLine <String> ] [-Comment <String>  
] [-DisableMomAlertOnRun <Boolean> ] [-DisableProgram <Boolean> ] [-DiskSpaceRequirement  
<String> ] [-DiskSpaceUnit {GB | KB | MB} ] [-DriveLetter <String> ] [-DriveMode  
{RenameWithUnc | RequiresDriveLetter | RequiresSpecificDriveLetter} ] [-Duration <Int32> ]  
[-EnableTaskSequence <Boolean> ] [-GenerateMomAlertOnFail <Boolean> ] [-ProgramAssignedType  
{RunOnceForEveryUserWhoLogsOn | RunOnceForTheComputer} ] [-ProgramRunType  
{OnlyWhenNoUserIsLoggedIn | OnlyWhenUserIsLoggedIn | WhetherOrNotUserIsLoggedIn} ] [-  
Reconnect <Boolean> ] [-Requirement <String> ] [-RunMode {RunWithAdministrativeRights |  
RunWithUserRights} ] [-RunType {Hidden | Maximized | Minimized | Normal} ] [-  
SuppressProgramNotifications <Boolean> ] [-UserInteraction <Boolean> ] [-WorkingDirectory  
<String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetDeviceProgramById

```
Set-CMProgram -DeviceProgramName <String> -Id <String> [-CommandLine <String> ] [-  
CommandLineFolder <String> ] [-Comment <String> ] [-DiskSpaceRequirement <String> ] [-  
DiskSpaceUnit {GB | KB | MB} ] [-DownloadProgramType {AsSoonAsPossible | OnlyOverFastNetwork  
| OnlyWhenTheDeviceIsDocked} ] [-Requirement <String> ] [-WorkingDirectory <String> ] [-  
Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetDeviceProgramByName

```
Set-CMProgram -DeviceProgramName <String> -Name <String> [-CommandLine <String> ] [-  
CommandLineFolder <String> ] [-Comment <String> ] [-DiskSpaceRequirement <String> ] [-  
DiskSpaceUnit {GB | KB | MB} ] [-DownloadProgramType {AsSoonAsPossible | OnlyOverFastNetwork  
| OnlyWhenTheDeviceIsDocked} ] [-Requirement <String> ] [-WorkingDirectory <String> ] [-  
Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetDeviceProgramByValue

```
Set-CMProgram -DeviceProgramName <String> -InputObject <IResultObject> [-CommandLine  
<String> ] [-CommandLineFolder <String> ] [-Comment <String> ] [-DiskSpaceRequirement  
<String> ] [-DiskSpaceUnit {GB | KB | MB} ] [-DownloadProgramType {AsSoonAsPossible |  
OnlyOverFastNetwork | OnlyWhenTheDeviceIsDocked} ] [-Requirement <String> ] [-  
WorkingDirectory <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetStandardProgramById

```
Set-CMProgram -Id <String> -StandardProgramName <String> [-AfterRunningType  
{ConfigurationManagerLogsUserOff | ConfigurationManagerRestartsComputer | NoActionRequired |  
ProgramControlsRestart} ] [-Category <String> ] [-CommandLine <String> ] [-Comment <String>
```

```

] [-DisableMomAlertOnRun <Boolean> ] [-DisableProgram <Boolean> ] [-DiskSpaceRequirement
<String> ] [-DiskSpaceUnit {GB | KB | MB} ] [-DriveLetter <String> ] [-DriveMode
{RenameWithUnc | RequiresDriveLetter | RequiresSpecificDriveLetter} ] [-Duration <Int32> ]
[-EnableTaskSequence <Boolean> ] [-GenerateMomAlertOnFail <Boolean> ] [-ProgramAssignedType
{RunOnceForEveryUserWhoLogsOn | RunOnceForTheComputer} ] [-ProgramRunType
{OnlyWhenNoUserIsLoggedIn | OnlyWhenUserIsLoggedIn | WhetherOrNotUserIsLoggedIn} ] [-
Reconnect <Boolean> ] [-Requirement <String> ] [-RunMode {RunWithAdministrativeRights |
RunWithUserRights} ] [-RunType {Hidden | Maximized | Minimized | Normal} ] [-
SuppressProgramNotifications <Boolean> ] [-UserInteraction <Boolean> ] [-WorkingDirectory
<String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]

```

Parameter Set: SetStandardProgramByValue

```

Set-CMProgram -InputObject <IResultObject> -StandardProgramName <String> [-AfterRunningType
{ConfigurationManagerLogsUserOff | ConfigurationManagerRestartsComputer | NoActionRequired |
ProgramControlsRestart} ] [-Category <String> ] [-CommandLine <String> ] [-Comment <String>
] [-DisableMomAlertOnRun <Boolean> ] [-DisableProgram <Boolean> ] [-DiskSpaceRequirement
<String> ] [-DiskSpaceUnit {GB | KB | MB} ] [-DriveLetter <String> ] [-DriveMode
{RenameWithUnc | RequiresDriveLetter | RequiresSpecificDriveLetter} ] [-Duration <Int32> ]
[-EnableTaskSequence <Boolean> ] [-GenerateMomAlertOnFail <Boolean> ] [-ProgramAssignedType
{RunOnceForEveryUserWhoLogsOn | RunOnceForTheComputer} ] [-ProgramRunType
{OnlyWhenNoUserIsLoggedIn | OnlyWhenUserIsLoggedIn | WhetherOrNotUserIsLoggedIn} ] [-
Reconnect <Boolean> ] [-Requirement <String> ] [-RunMode {RunWithAdministrativeRights |
RunWithUserRights} ] [-RunType {Hidden | Maximized | Minimized | Normal} ] [-
SuppressProgramNotifications <Boolean> ] [-UserInteraction <Boolean> ] [-WorkingDirectory
<String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]

```

Detailed Description

The **Set-CMProgram** cmdlet modifies a program in Microsoft System Center 2012 Configuration Manager. Programs are commands that are associated with a System Center 2012 Configuration Manager package. Programs identify the actions that occur when the client receives the client package. You can associate multiple programs with the same package.

Parameters

-AfterRunningType<AfterRunningType>

Specifies the action that occurs after the program completes successfully. Valid values are: ConfigurationManagerLogUserOff, ConfigurationManagerStartsComputer, NoActionRequired, and ProgramControlsRestart.

The acceptable values for this parameter are:

ConfigurationManagerLogsUserOff	
ConfigurationManagerRestartsComputer	

NoActionRequired	
ProgramControlsRestart	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Category<String>

Specifies the category under which the program is displayed on the client computer.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CommandLine<String>

Specifies the command line for the program.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CommandLineFolder<String>

Specifies the folder that contains the executable program. This folder can be an absolute path on the client, or a path relative to the distribution folder that contains the package.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Comment<String>

Specifies optional text about a program, such as a description. On client computers, this text appears in Run Advertised Programs in Control Panel.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeviceProgramName<String>

Specifies a device program name.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DisableMomAlertOnRun<Boolean>

Indicates whether the computer running the program is in maintenance mode for the duration of the program. When in maintenance mode, Microsoft Operations Manager (MOM) disables alerts while the program runs.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DisableProgram<Boolean>

Indicates whether to temporarily disable all advertisements that contain this program. If this option is selected, the program is removed from the list of available programs for users to run and it will not be run through assignment until re-enabled.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DiskSpaceRequirement<String>

Specifies the amount of disk space that the software program requires to run on the computer. Requires the *DiskSpaceUnit* parameter be set. The value must be greater than or equal to zero.

Aliases	none
Required?	false
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DiskSpaceUnit<DiskSpaceUnitType>

Specifies the units, GB, KB, or MB, for the *DiskSpaceRequirement* parameter.

The acceptable values for this parameter are:

GB	
KB	
MB	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DownloadProgramType<DownloadProgramType>

Specifies when the program is to run. Valid values are: *AsSoonAsPossible*, *OnlyOverFastNetwork*, and *OnlyWhenTheDeviceIsLocked*.

The acceptable values for this parameter are:

<i>AsSoonAsPossible</i>	
<i>OnlyOverFastNetwork</i>	
<i>OnlyWhenTheDeviceIsDocked</i>	

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DriveLetter<String>

Specifies a drive letter to qualify the location if the *DriveMode* parameter is used.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DriveMode<DriveModeType>

Indicates whether the program requires a specific drive letter, specified in the *DriveLetter* parameter. By default, the program runs with a Universal Naming Convention (UNC) name. If *DriveMode* is set to *RequiresDriveLetter*, the program uses any available drive letter. If *DriveMode* is set to *RequiresSpecificDriveLetter*, the program only runs if the drive is not already used.

The acceptable values for this parameter are:

RenameWithUnc	
RequiresDriveLetter	
RequiresSpecificDriveLetter	

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Duration<Int32>

Specifies the maximum amount of time the program is expected to run. The default value is 120 minutes.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnableTaskSequence<Boolean>

Indicates whether this program can be installed from the Install Software task sequence.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-GenerateMomAlertOnFail<Boolean>

Indicates whether MOM generates an application log event entry if the program fails.

Aliases	none
Required?	false
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Id<String>

Specifies an ID for the program.

Aliases	Packageld
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a **CMProgram** object. To obtain a **CMProgram** object, use the [Get-CMProgram](#) cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies a name for the program.

Aliases	none
Required?	true
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ProgramAssignedType<ProgramAssignedType>

Specifies whether the program runs once on the computer, or once for every user who logs on to the computer. The default setting is RunOnceForTheComputer. The program is only assigned to users when the *ProgramRunType* parameter is set to OnlyWhenUserIsLoggedIn.

The acceptable values for this parameter are:

RunOnceForEveryUserWhoLogsOn	
RunOnceForTheComputer	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ProgramRunType<ProgramRunType>

Specifies the logon conditions that are necessary for the program to run. Valid values are: OnlyWhenNoUserIsLoggedIn, OnlyWhenUserIsLoggedIn, and WhetherOrNotUserIsLoggedIn. The default setting is OnlyWhenUserIsLoggedIn.

The acceptable values for this parameter are:

OnlyWhenNoUserIsLoggedIn	
OnlyWhenUserIsLoggedIn	
WhetherOrNotUserIsLoggedIn	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Reconnect<Boolean>

Indicates whether the client computer reconnects to the distribution point when the user logs on.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Requirement<String>

Specifies any additional requirements for standard or device programs.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RunMode<RunModeType>

Specifies the credentials the client computer requires to run the program, either RunWithAdministrativeRights or RunWithUserRights.

The acceptable values for this parameter are:

RunWithAdministrativeRights	
RunWithUserRights	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RunType<RunType>

Specifies the mode in which the program will run on the client computer. Valid values are: Hidden, Maximized, Minimized, and Normal. The default is Normal.

The acceptable values for this parameter are:

Hidden	
Maximized	
Minimized	
Normal	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-StandardProgramName<String>

Specifies the standard program name.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SuppressProgramNotifications<Boolean>

Indicates whether to display notification area icons and messages, as well as countdown notifications, for this program.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UserInteraction<Boolean>

Indicates whether to allow users to interact with the program.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-WorkingDirectory<String>

Specifies a working directory for the program.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Modify a standard program

This command modifies a standard program.

```
PS C:\> Set-CMProgram -Name "Test" -StandardProgramName SPM -Comment "Standard Upgrades" -  
CommandLine "RunThisNow" -RunType Maximized -AfterRunningType ProgramControlsRestart -  
Category "Laptops" -DiskSpaceRequirement 50 -DiskSpaceUnit MB -Duration 150 -Requirement 4 -  
Reconnect $False -SuppressProgramNotifications $False -DisableProgram $True -  
EnableTaskSequence $True -DisableMomAlertOnRun $True -GenerateMomAlertOnFail $True
```

Example 2: Modify a device program

This command modifies a device program.

```
PS C:\> Set-CMProgram -Name "Test" -DeviceProgramName DPM -Comment "Upgrades for December" -  
CommandLine "RunMe" -WorkingDirectory "\TempWork" -CommandLineFolder "C:\Windows" -  
DiskSpaceRequirement 30 -DiskSpaceUnit MB -DownloadProgramType AsSoonAsPossible -Requirement  
"All previous device updates"
```

Related topics

[Disable-CMProgram](#)

[Enable-CMProgram](#)

[Get-CMProgram](#)

[New-CMProgram](#)

[Remove-CMProgram](#)

Set-CMQueryResultMaximum

Set-CMQueryResultMaximum

Changes the setting for the query result maximum.

Syntax

Parameter Set: Value

```
Set-CMQueryResultMaximum [-Maximum] <Int32> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Set-CMQueryResultMaximum** cmdlet changes the setting for the maximum number of rows that a Microsoft System Center 2012 Configuration Manager report query can return.

Parameters

-Maximum<Int32>

Specifies the maximum number of rows that a Configuration Manager report query can return. The default value is 10,000.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Set the query result maximum

This command sets the maximum number of rows that a Configuration Manager report query can return to 2500.

```
PS C:\> Set-CMQueryResultMaximum -Maximum 2500
```

Related topics

[Get-CMQueryResultMaximum](#)

Set-CMRemoteConnectionProfileConfigurationItem

Set-CMRemoteConnectionProfileConfigurationItem

Modifies a remote connection profile.

Syntax

Parameter Set: SetByName

```
Set-CMRemoteConnectionProfileConfigurationItem -Name <String[]> [-Description <String> ] [-EnableNLA <Boolean> ] [-EnablePrimaryUsers <Boolean> ] [-EnableTSConnection <Boolean> ] [-EnableTSFirewallRule <Boolean> ] [-NewName <String> ] [-RDGatewayServer <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetById

```
Set-CMRemoteConnectionProfileConfigurationItem -Id <String[]> [-Description <String> ] [-EnableNLA <Boolean> ] [-EnablePrimaryUsers <Boolean> ] [-EnableTSConnection <Boolean> ] [-EnableTSFirewallRule <Boolean> ] [-NewName <String> ] [-RDGatewayServer <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetByValue

```
Set-CMRemoteConnectionProfileConfigurationItem -InputObject <IResultObject> [-Description <String> ] [-EnableNLA <Boolean> ] [-EnablePrimaryUsers <Boolean> ] [-EnableTSConnection <Boolean> ] [-EnableTSFirewallRule <Boolean> ] [-NewName <String> ] [-RDGatewayServer <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Set-CMRemoteConnectionProfileConfigurationItem** cmdlet modifies a remote connection profile. Client computers use remote connection profiles to remotely connect to computers from outside the domain or over the Internet.

Parameters

-Description<String>

Specifies a description for a remote connection profile.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnableNLA<Boolean>

Indicates whether to allow connections only from computers that run Remote Desktop by using Network Level Authentication.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnablePrimaryUsers<Boolean>

Indicates whether all primary users of the client computer are allowed to remotely connect. If you specify a value for this parameter, you must specify values for the *EnableTSConnection* and *EnableTSFirewallRule* parameters.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnableTSConnection<Boolean>

Indicates whether to allow remote connections to client computers. If you specify a value for this parameter, you must specify values for the *EnablePrimaryUsers* and *EnableTSFirewallRule* parameters.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnableTSFirewallRule<Boolean>

Indicates whether to allow Windows Firewall exceptions for connections in Windows domains and on private networks. If you specify a value for this parameter, you must specify values for the *EnablePrimaryUsers* and *EnableTSConnections* parameters.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Id<String[]>

Specifies an array of IDs for remote connection profiles.

Aliases	CId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-InputObject<IResultObject>

Specifies a remote connection profile object. To obtain a remote connection profile, use the **Get-CMRemoteConnectionProfileConfigurationItem** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String[]>

Specifies an array of names of remote connection profiles.

Aliases	LocalizedDisplayName
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-NewName<String>

Specifies the new name for the remote connection profile.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RDGatewayServer<String>

Specifies the host name and port of the Remote Desktop gateway server, for example, Boston.gateway.Contoso.com:8080.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

[Copy-CMRemoteConnectionProfileConfigurationItem](#)

[Get-CMRemoteConnectionProfileConfigurationItem](#)

[New-CMRemoteConnectionProfileConfigurationItem](#)

[Remove-CMRemoteConnectionProfileConfigurationItem](#)

Set-CMReportingServicePoint

Set-CMReportingServicePoint

Modifies a System Center 2012 Configuration Manager reporting service point.

Syntax

Parameter Set: SetByName

```
Set-CMReportingServicePoint -SiteCode <String> -SiteSystemServerName <String> [-DatabaseName <String> ] [-DatabaseServerName <String> ] [-UserName <String> ] [-Confirm] [-WhatIf] [<CommonParameters>]
```

Parameter Set: SetByValue

```
Set-CMReportingServicePoint -InputObject <IResultObject> [-DatabaseName <String> ] [-DatabaseServerName <String> ] [-UserName <String> ] [-Confirm] [-WhatIf] [<CommonParameters>]
```

Detailed Description

The **Set-CMReportingServicePoint** cmdlet modifies a Microsoft System Center 2012 Configuration Manager reporting service point. A reporting service point is a site system role that is installed on a server that is running Microsoft SQL Server Reporting Services.

Parameters

-DatabaseName<String>

Specifies the name of the Configuration Manager database that you want to use as the data source for reports from Microsoft SQL Server Reporting Services.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DatabaseServerName<String>

Specifies the name of the Configuration Manager database server that you want to use as the data source for the reports from Microsoft SQL Server Reporting Services. To specify a database instance, use the format <Server Name>\<Instance Name>.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies an input object. To obtain an input object, use the **Get-CMReportingServicePoint** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteCode<String>

Specifies a site code of a Configuration Manager site that hosts this system role.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteSystemServerName<String>

Specifies the name of a server hosting the site system role.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UserName<String>

Specifies a user name that Configuration Manager uses to connect with SQL Server Reporting Services and that gives this user access to the site database.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Set a reporting service point

The command sets a reporting service point by using the *SiteSystemServerName* parameter.

```
PS C:\> Set-CMReportingServicePoint -SiteSystemServerName "Contoso-Test.Contoso.Com" -  
SiteCode "CM4" -UserName "Contoso\DavidChew"
```

Example 2: Set a reporting service point by using a site system server name

The command sets a reporting service point by using the *SiteSystemServerName* parameter.

```
PS C:\> Set-CMReportingServicePoint -SiteSystemServerName "Contoso-Test.Contoso.Com" -  
SiteCode "CM4" -DatabaseServerName "Contoso-TestDB.Contoso.Com" -DatabaseName "CM_CM2"
```

Example 3: Set a reporting service point by using an input object

The first command uses the **Get-CMReportingServicePoint** cmdlet to get a reporting service point, using the *SiteSystemServerName* parameter.

The second command uses the **Set-CMReportingServicePoint** cmdlet to set the reporting point by using the input object.

```
PS C:\> $RS = Get-CMReportingServicePoint -SiteSystemServerName "Contoso-Test.Contoso.Com" -  
SiteCode "CM4"
```

```
PS C:\> Set-CMReportingServicePoint -InputObject $RS -DatabaseServerName "Contoso-  
TestDB.Contoso.Com" -DatabaseName "CM_CM4"
```

Related topics

[Add-CMReportingServicePoint](#)

[Get-CMReportingServicePoint](#)

[Remove-CMReportingServicePoint](#)

Set-CMSecurityRole

Set-CMSecurityRole

Changes configuration settings of a security role.

Syntax

Parameter Set: SetById

```
Set-CMSecurityRole -Id <String[]> [-Description <String> ] [-NewName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetByName

```
Set-CMSecurityRole -Name <String[]> [-Description <String> ] [-NewName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetValue

```
Set-CMSecurityRole -InputObject <IResultObject> [-Description <String> ] [-NewName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Set-CMSecurityRole** cmdlet changes configuration settings of a security role. You can use this cmdlet to change the name and description of a security role.

Parameters

-Description<String>

Specifies the description of a security role.

Aliases	RoleDescription
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Id<String[]>

Specifies an array of IDs of security roles.

Aliases	RoleId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a **CMSecurityRole** object. To obtain a **CMSecurityRole** object, use the **Get-
CMSecurityRole** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String[]>

Specifies an array of names of security roles.

Aliases	RoleName
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	true

-NewName<String>

Specifies a new name for the security role.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Change the name of a security role by using an ID

This command renames the security role that has the ID CM100003. The command changes the name to RTOperator02.

```
PS C:\> Set-CMSecurityRole -Id "CM100003" -NewName "RTOperator02"
```

Example 2: Change the name of a security role by using a name

This command renames the security role named SRole02. The command changes the name to RTOperator02.

```
PS C:\> Set-CMSecurityRole -Name "SRole02" -NewName "RTOperator02"
```

Example 3: Change the name of a security role by using an object variable

The first command gets the security role that has the ID CM100003 and stores it in the \$Srole variable. The second command renames the security role for the object stored in \$Srole. The command changes the name to RTOperator02.

```
PS C:\> $Srole = Get-CMSecurityRole -Id "CM100003"  
PS C:\> Set-CMSecurityRole -Inputobject $Srole -NewName "RTOperator02"
```

Related topics

[Get-CMSecurityRole](#)

[Copy-CMSecurityRole](#)

[Remove-CMSecurityRole](#)

[Import-CMSecurityRole](#)

[Export-CMSecurityRole](#)

[Remove-CMSecurityRoleFromAdministrativeUser](#)

Set-CMSecurityScope

Set-CMSecurityScope

Changes configuration settings of a security scope.

Syntax

Parameter Set: SetById

```
Set-CMSecurityScope -Id <String[]> [-Description <String> ] [-NewName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetByName

```
Set-CMSecurityScope -Name <String[]> [-Description <String> ] [-NewName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetValue

```
Set-CMSecurityScope -InputObject <IResultObject> [-Description <String> ] [-NewName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Set-CMSecurityScope** cmdlet changes configuration settings of a security scope. You can use this cmdlet to change the name and description of a security scope.

Parameters

-Description<String>

Specifies a description of a security scope.

Aliases	CategoryDescription
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Id<String[]>

Specifies an array of IDs of security scopes.

Aliases	CategoryId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a **CMSecurityScope** object. To obtain a **CMSecurityScope** object, use the **Get-
CMSecurityScope** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String[]>

Specifies an array of names of security scopes.

Aliases	CategoryName
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	true

-NewName<String>

Specifies a new name for the security scope.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Change the settings of a security scope by using an ID

This command renames the security scope that has the ID CM200001. The command changes the name to ScopeT03 and adds a description to the security scope.

```
PS C:\> Set-CMSecurityScope -Id "CM200001" -NewName "ScopeT03" -Description "Security scope for team 3."
```

Example 2: Change the settings of a security scope by using a name

This command renames the security scope named ScopeT01. The command changes the name to ScopeT03 and adds a description to the security scope.

```
PS C:\> Set-CMSecurityScope -Name "ScopeT01" -NewName "ScopeT03" -Description "Security scope for team 3."
```

Example 3: Change the settings of a security scope by using an object variable

The first command gets the security scope that has the ID CM200001 and stores it in the \$Sscope variable.

The second command renames the security scope for the object stored in \$Sscope. The command changes the name to ScopeT03 and adds a description to the security scope.

```
PS C:\> $Sscope = Get-CMSecurityScope -Id "CM200001"  
PS C:\> Set-CMSecurityScope -Inputobject $Sscope -NewName "ScopeT03" -Description "Security scope for team 3."
```

Related topics

[Get-CMSecurityScope](#)

[New-CMSecurityScope](#)

[Remove-CMSecurityScope](#)

[Remove-CMSecurityScopeFromAdministrativeUser](#)

Set-CMSite

Set-CMSite

Changes security scope settings for Configuration Manager sites.

Syntax

Parameter Set: SetByNameMandatory

```
Set-CMSite -Name <String> [-AddActiveDirectoryForest <IResultObject[]> ] [-  
AddCertificateByPath <String[]> ] [-AddClientRequestServiceType {ClientNotificationTcp |  
ClientRequestHttpTcp | ClientRequestHttpTcpDefault | ClientRequestsHttpsTcp |  
ClientRequestsHttpsTcpDefault | WakeOnLanUdp} ] [-ClientCertificateCustomStoreName <String>  
] [-ClientCertificateSelectionCriteriaType {CertificateSubjectContainsString |  
CertificateSubjectOrSanIncludesAttributes | ClientAuthentication} ] [-  
ClientCertificateSelectionCriteriaValue <String> ] [-  
ClientCheckCertificateRevocationListForSiteSystem <Boolean> ] [-  
ClientComputerCommunicationType {HttpsOnly | HttpsOrHttp} ] [-Comment <String> ] [-  
ConcurrentSendingDelayBeforeRetryingMinutes <Int32> ] [-  
CriticalAlertWhenFreeDiskSpaceFallBelowFollowingValueGB <Int32> ] [-EnableWakeOnLan  
<Boolean> ] [-GenerateAlertWhenFreeDiskSpaceOnSiteDatabaseIsLow <Boolean> ] [-  
MaximumConcurrentSendingForAllSite <Int32> ] [-MaximumConcurrentSendingForPerSite <Int32> ]  
[-MaximumNumberOfSendingWakeUpPacketBeforePausing <Int32> ] [-  
PortForClientRequestServiceType <Int32> ] [-RemoveActiveDirectoryForest <IResultObject[]> ]  
[-RemoveCertificateByKey <String[]> ] [-RemoveClientRequestServiceType  
{ClientNotificationTcp | ClientRequestHttpTcp | ClientRequestHttpTcpDefault |  
ClientRequestsHttpsTcp | ClientRequestsHttpsTcpDefault | WakeOnLanUdp} ] [-RequireSha256  
<Boolean> ] [-RequireSigning <Boolean> ] [-RetryNumberForConcurrentSending <Int32> ] [-  
RetryNumberOfSendingWakeUpPacketTransmission <Int32> ] [-  
SendingWakeUpPacketBeforePausingWaitSeconds <Int32> ] [-  
SendingWakeUpPacketTransmissionDelayMinutes <Int32> ] [-  
SendingWakeUpPacketTransmissionOffsetMinutes <Int32> ] [-  
TakeActionForMultipleCertificateMatchCriteria {FailSelectionAndSendErrorMessage |  
SelectCertificateWithLongestValidityPeriod} ] [-ThreadNumberOfSendingWakeUpPacket <Int32> ]  
[-UseCustomWebSite <Boolean> ] [-UseEncryption <Boolean> ] [-UsePkiClientCertificate  
<Boolean> ] [-WakeOnLanTransmissionMethodType {SubnetDirectedBroadcasts | Unicast} ] [-  
WakeOnLanType {UseAmtPowerOnCommandsOnly | UseAmtPowerOnCommandsORWakeUpPackets |  
UseWakeUpPacketsOnly} ] [-WarningAlertWhenFreeDiskSpaceFallBelowFollowingValueGB <Int32> ]  
[-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetByObject

```
Set-CMSite -InputObject <IResultObject> [-AddActiveDirectoryForest <IResultObject[]> ] [-  
AddCertificateByPath <String[]> ] [-AddClientRequestServiceType {ClientNotificationTcp |  
ClientRequestHttpTcp | ClientRequestHttpTcpDefault | ClientRequestsHttpsTcp |  
ClientRequestsHttpsTcpDefault | WakeOnLanUdp} ] [-ClientCertificateCustomStoreName <String>  
] [-ClientCertificateSelectionCriteriaType {CertificateSubjectContainsString |  
CertificateSubjectOrSanIncludesAttributes | ClientAuthentication} ] [-  
ClientCertificateSelectionCriteriaValue <String> ] [-
```

```
ClientCheckCertificateRevocationListForSiteSystem <Boolean> ] [-
ClientComputerCommunicationType {HttpsOnly | HttpsOrHttp} ] [-Comment <String> ] [-
ConcurrentSendingDelayBeforeRetryingMinutes <Int32> ] [-
CriticalAlertWhenFreeDiskSpaceFallBelowFollowingValueGB <Int32> ] [-EnableWakeOnLan
<Boolean> ] [-GenerateAlertWhenFreeDiskSpaceOnSiteDatabaseIsLow <Boolean> ] [-
MaximumConcurrentSendingForAllSite <Int32> ] [-MaximumConcurrentSendingForPerSite <Int32> ]
[-MaximumNumberOfSendingWakeUpPacketBeforePausing <Int32> ] [-
PortForClientRequestServiceType <Int32> ] [-RemoveActiveDirectoryForest <IResultObject[]> ]
[-RemoveCertificateByKey <String[]> ] [-RemoveClientRequestServiceType
{ClientNotificationTcp | ClientRequestHttpTcp | ClientRequestHttpTcpDefault |
ClientRequestsHttpsTcp | ClientRequestsHttpsTcpDefault | WakeOnLanUdp} ] [-RequireSha256
<Boolean> ] [-RequireSigning <Boolean> ] [-RetryNumberForConcurrentSending <Int32> ] [-
RetryNumberOfSendingWakeUpPacketTransmission <Int32> ] [-
SendingWakeUpPacketBeforePausingWaitSeconds <Int32> ] [-
SendingWakeUpPacketTransmissionDelayMinutes <Int32> ] [-
SendingWakeUpPacketTransmissionOffsetMinutes <Int32> ] [-
TakeActionForMultipleCertificateMatchCriteria {FailSelectionAndSendErrorMessage |
SelectCertificateWithLongestValidityPeriod} ] [-ThreadNumberOfSendingWakeUpPacket <Int32> ]
[-UseCustomWebSite <Boolean> ] [-UseEncryption <Boolean> ] [-UsePkiClientCertificate
<Boolean> ] [-WakeOnLanTransmissionMethodType {SubnetDirectedBroadcasts | Unicast} ] [-
WakeOnLanType {UseAmtPowerOnCommandsOnly | UseAmtPowerOnCommandsORWakeUpPackets |
UseWakeUpPacketsOnly} ] [-WarningAlertWhenFreeDiskSpaceFallBelowFollowingValueGB <Int32> ]
[-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetBySiteCodeMandatory

```
Set-CMSite -SiteCode <String> [-AddActiveDirectoryForest <IResultObject[]> ] [-
AddCertificateByPath <String[]> ] [-AddClientRequestServiceType {ClientNotificationTcp |
ClientRequestHttpTcp | ClientRequestHttpTcpDefault | ClientRequestsHttpsTcp |
ClientRequestsHttpsTcpDefault | WakeOnLanUdp} ] [-ClientCertificateCustomStoreName <String>
] [-ClientCertificateSelectionCriteriaType {CertificateSubjectContainsString |
CertificateSubjectOrSanIncludesAttributes | ClientAuthentication} ] [-
ClientCertificateSelectionCriteriaValue <String> ] [-
ClientCheckCertificateRevocationListForSiteSystem <Boolean> ] [-
ClientComputerCommunicationType {HttpsOnly | HttpsOrHttp} ] [-Comment <String> ] [-
ConcurrentSendingDelayBeforeRetryingMinutes <Int32> ] [-
CriticalAlertWhenFreeDiskSpaceFallBelowFollowingValueGB <Int32> ] [-EnableWakeOnLan
<Boolean> ] [-GenerateAlertWhenFreeDiskSpaceOnSiteDatabaseIsLow <Boolean> ] [-
MaximumConcurrentSendingForAllSite <Int32> ] [-MaximumConcurrentSendingForPerSite <Int32> ]
[-MaximumNumberOfSendingWakeUpPacketBeforePausing <Int32> ] [-
PortForClientRequestServiceType <Int32> ] [-RemoveActiveDirectoryForest <IResultObject[]> ]
[-RemoveCertificateByKey <String[]> ] [-RemoveClientRequestServiceType
{ClientNotificationTcp | ClientRequestHttpTcp | ClientRequestHttpTcpDefault |
ClientRequestsHttpsTcp | ClientRequestsHttpsTcpDefault | WakeOnLanUdp} ] [-RequireSha256
<Boolean> ] [-RequireSigning <Boolean> ] [-RetryNumberForConcurrentSending <Int32> ] [-
RetryNumberOfSendingWakeUpPacketTransmission <Int32> ] [-
SendingWakeUpPacketBeforePausingWaitSeconds <Int32> ] [-
SendingWakeUpPacketTransmissionDelayMinutes <Int32> ] [-
SendingWakeUpPacketTransmissionOffsetMinutes <Int32> ] [-
TakeActionForMultipleCertificateMatchCriteria {FailSelectionAndSendErrorMessage |
SelectCertificateWithLongestValidityPeriod} ] [-ThreadNumberOfSendingWakeUpPacket <Int32> ]
[-UseCustomWebSite <Boolean> ] [-UseEncryption <Boolean> ] [-UsePkiClientCertificate
<Boolean> ] [-WakeOnLanTransmissionMethodType {SubnetDirectedBroadcasts | Unicast} ] [-
```

```
WakeOnLanType {UseAmtPowerOnCommandsOnly | UseAmtPowerOnCommandsORWakeUpPackets |
UseWakeUpPacketsOnly} ] [-WarningAlertWhenFreeDiskSpaceFallBelowFollowingValueGB <Int32> ]
[-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetSecurityScopeByName

```
Set-CMSite -Name <String> -SecurityScopeAction {AddMembership | RemoveMembership} -
SecurityScopeName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetSecurityScopeBySiteCode

```
Set-CMSite -SecurityScopeAction {AddMembership | RemoveMembership} -SecurityScopeName
<String> -SiteCode <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetSecurityScopeByValue

```
Set-CMSite -InputObject <IResultObject> -SecurityScopeAction {AddMembership |
RemoveMembership} -SecurityScopeName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Set-CMSite** cmdlet changes security scope settings for one or more Microsoft System Center 2012 Configuration Manager sites. A security scope is a collection of permissions that, in conjunction with security roles, defines the configuration actions that an administrator can perform on the site. You can use this cmdlet to change the type of a security scope action and the name of a security scope for a System Center 2012 Configuration Manager site. You can specify a site for which you change security scope settings by using a site name or a site code, or you can use the **Get-CMSite** cmdlet to specify a site.

Parameters

-AddActiveDirectoryForest<IResultObject[]>

Specifies an array of Active Directory Forest objects to publish in Active Directory Domain Services. To obtain an AD Forest object, use the **Get-ADForest** cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AddCertificateByPath<String[]>

Specifies an array of paths to certificates.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AddClientRequestServiceType<ClientRequestServiceType>

Specifies a service type to add. Valid values are:

- ClientNotificationTcp
- ClientRequestHttpTcp
- ClientRequestHttpTcpDefault
- ClientRequestHttpsTcp
- ClientRequestHttpsTcpDefault
- WakeOnLanUdp

The acceptable values for this parameter are:

ClientNotificationTcp	
ClientRequestHttpTcp	
ClientRequestHttpTcpDefault	
ClientRequestsHttpsTcp	
ClientRequestsHttpsTcpDefault	
WakeOnLanUdp	

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ClientCertificateCustomStoreName<String>

Specifies the name of a custom store that contains client certificates.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-

ClientCertificateSelectionCriteriaType<ClientCertificateSelectionCriteriaType>

Specifies the criteria type to match in a client certificate, such as a string or attribute for a subject or subject alt name.

The acceptable values for this parameter are:

CertificateSubjectContainsString	
CertificateSubjectOrSanIncludesAttributes	
ClientAuthentication	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ClientCertificateSelectionCriteriaValue<String>

Specifies a value for the *ClientCertificateSelectionCriteriaType*.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ClientCheckCertificateRevocationListForSiteSystem<Boolean>

Indicates whether to check the Certificate Revocation List (CRL) for a certificate.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-

ClientComputerCommunicationType<ClientComputerCommunicationType>

Specifies the communication type. Valid values are: `HttpsOnly` and `HttpsOrHttp`.

The acceptable values for this parameter are:

HttpsOnly	
HttpsOrHttp	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Comment<String>

Specifies a comment for a Configuration Manager site.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ConcurrentSendingDelayBeforeRetryingMinutes<Int32>

Specifies the number of minutes to wait before retrying failed communication.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-

CriticalAlertWhenFreeDiskSpaceFallBelowFollowingValueGB<Int32>

Specifies the free disk space threshold, in gigabytes, for a critical alert.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnableWakeOnLan<Boolean>

Indicates whether to send Wake On LAN packets for scheduled activities, such as software update deployment.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-GenerateAlertWhenFreeDiskSpaceOnSiteDatabasesLow<Boolean>

Indicates whether to generate an alert when free disk space on the site database reaches a threshold.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a Configuration Manager site object. To obtain a Configuration Manager site object, use the **Get-CMSite** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-MaximumConcurrentSendingForAllSite<Int32>

Specifies the maximum number of simultaneous communications to all sites.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-MaximumConcurrentSendingForPerSite<Int32>

Specifies the maximum number of simultaneous communications to any single site.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-MaximumNumberOfSendingWakeUpPacketBeforePausing<Int32>

Specifies the maximum number of wake up packets transmitted by this site server before pausing.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies the name of a Configuration Manager site.

Aliases	SiteName
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PortForClientRequestServiceType<Int32>

Specifies a port number, such as 80 or 8080, for client requests.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RemoveActiveDirectoryForest<IResultObject[]>

Specifies an array of Active Directory Forest objects to remove from Active Directory Domain Services.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RemoveCertificateByKey<String[]>

Specifies an array of certificates to remove.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RemoveClientRequestServiceType<ClientRequestServiceType>

Specifies a service type to remove. Valid values are:

- ClientNotificationTcp
- ClientRequestHttpTcp
- ClientRequestHttpTcpDefault
- ClientRequestHttpsTcp
- ClientRequestHttpsTcpDefault
- WakeOnLanUdp

The acceptable values for this parameter are:

ClientNotificationTcp	
ClientRequestHttpTcp	

ClientRequestHttpTcpDefault	
ClientRequestsHttpsTcp	
ClientRequestsHttpsTcpDefault	
WakeOnLanUdp	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RequireSha256<Boolean>

Indicates whether to use the SHA-256 algorithm to sign communications.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RequireSigning<Boolean>

Indicates whether to require Configuration Manager sites to sign communications with other sites.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RetryNumberForConcurrentSending<Int32>

Specifies the number of times to retry a failed communication.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RetryNumberOfSendingWakeUpPacketTransmission<Int32>

Specifies the number of times a wake up packet is sent to a target computer.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SecurityScopeAction<SecurityScopeActionType>

Specifies an action for a security scope. Valid values are: AddMembership and RemoveMembership.

The acceptable values for this parameter are:

AddMembership	
RemoveMembership	

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SecurityScopeName<String>

Specifies the name of a security scope. A security scope name can be Default or the name of a custom security scope.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SendingWakeUpPacketBeforePausingWaitSeconds<Int32>

Specifies the number of wake up packets to send before pausing. The default value is 10.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SendingWakeUpPacketTransmissionDelayMinutes<Int32>

Specifies the amount of time that wake up packet transmission is halted between retries.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SendingWakeUpPacketTransmissionOffsetMinutes<Int32>

Specifies the amount of time that wake up packets are sent prior to a scheduled activity that is enabled for Wake On LAN. The default value is 3.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteCode<String>

Specifies a site code for a Configuration Manager site to which you assign security scopes.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-

TakeActionForMultipleCertificateMatchCriteria<TakeActionForMultipleCertificateMatchCriteria>

Specifies the action to take for multiple matches of certificate criteria.

The acceptable values for this parameter are:

FailSelectionAndSendErrorMessage	
SelectCertificateWithLongestValidityPeriod	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ThreadNumberOfSendingWakeUpPacket<Int32>

Specifies the number of threads a site server uses when sending wake up packets.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UseCustomWebSite<Boolean>

Indicates whether to use a custom web site. Use a custom web site when you do not want to use the default web site.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UseEncryption<Boolean>

Indicates whether to use encryption for communication between sites.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UsePkiClientCertificate<Boolean>

Indicates whether to use a PKI certificate management solution.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-

WakeOnLanTransmissionMethodType<WakeOnLanTransmissionMethodType>

Specifies the type of transmission method to use for Wake On LAN transmissions.

The acceptable values for this parameter are:

SubnetDirectedBroadcasts	
Unicast	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-WakeOnLanType<WakeOnLanType>

Specifies the type of Wake On LAN packet to use.

The acceptable values for this parameter are:

UseAmtPowerOnCommandsOnly	
UseAmtPowerOnCommandsORWakeUpPackets	
UseWakeUpPacketsOnly	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-

WarningAlertWhenFreeDiskSpaceFallBelowFollowingValueGB<Int32

>

Specifies a threshold, in gigabytes, for free disk space.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Add a site to a security scope by using a site name

This command assigns a custom security scope named Scope22 to a System Center 2012 Configuration Manager site named CMSiteSystem.

```
PS C:\> Set-CMSite -SecurityScopeAction AddMembership -SecurityScopeName "Scope22" -SiteName "CMSiteSystem"
```

Example 2: Remove a security scope for a site by using the site name

This command removes the custom security scope in the previous example from a System Center 2012 Configuration Manager site named CMSiteSystem.

```
PS C:\> Set-CMSite -SecurityScopeAction RemoveMembership -SecurityScopeName "Scope22" -SiteName "CMSiteSystem"
```

Related topics

[Get-CMSite](#)

Set-CMSiteMaintenanceTask

Set-CMSiteMaintenanceTask

Changes settings for a Configuration Manager maintenance task.

Syntax

Parameter Set: SetMaintenanceTasksByName

```
Set-CMSiteMaintenanceTask -MaintenanceTask {BackupSiteServer |  
CheckApplicationTitleWithInventoryInformation | ClearUndiscoveredClients |  
DeleteAgedApplicationRequestData | DeleteAgedClientOperations |  
DeleteAgedClientPresenceHistory | DeleteAgedCollectedFiles |  
DeleteAgedComputerAssociationData | DeleteAgedDeleteDetectionData |  
DeleteAgedDevicesManagedByTheExchangeServerConnector | DeleteAgedDeviceWipeRecord |  
DeleteAgedDiscoveryData | DeleteAgedDistributionPointUsageStats |  
DeleteAgedEndpointProtectionHealthStatusHistoryData | DeleteAgedEnrolledDevices |  
DeleteAgedInventoryHistory | DeleteAgedLogData | DeleteAgedNotificationTaskHistory |  
DeleteAgedReplicationSummaryData | DeleteAgedReplicationTrackingData |  
DeleteAgedSoftwareMeteringData | DeleteAgedSoftwareMeteringSummaryData |  
DeleteAgedStatusMessages | DeleteAgedThreatData | DeleteAgedUnknownComputers |  
DeleteAgedUserDeviceAffinityData | DeleteInactiveClientDiscoveryData | DeleteObsoleteAlerts  
| DeleteObsoleteClientDiscoveryData | DeleteObsoleteForestDiscoverySitesAndSubnets |  
DeleteUnusedApplicationRevisions | EvaluateProvisionedAmtComputerCertificates | MonitorKeys  
| RebuildIndexes | SummarizeInstalledSoftwareData | SummarizeSoftwareMeteringFileUsageData |  
SummarizeSoftwareMeteringMonthlyUsageData} -SiteCode <String> [-BeginTime <DateTime> ] [-  
DaysOfWeek {Friday | Monday | Saturday | Sunday | Thursday | Tuesday | Wednesday} ] [-  
DeleteThanOlderDays <Int32> ] [-DeviceName <String> ] [-Enabled <Boolean> ] [-EnabledAlert  
<Boolean> ] [-LatestBeginTime <DateTime> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetSummaryTaskByName

```
Set-CMSiteMaintenanceTask -SummaryTask {UpdateApplicationCatalogTables} [-Enabled <Boolean>  
] [-RunIntervalMinutes <Int32> ] [-RunNow <Boolean> ] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Detailed Description

The **Set-CMSiteMaintenanceTask** cmdlet changes settings for a Microsoft System Center 2012 Configuration Manager maintenance task.

Parameters

-BeginTime<DateTime>

Specifies the date and time at which a maintenance task starts.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DaysOfWeek<DaysOfWeek[]>

Specifies an array of day names that determine the days of each week on which the maintenance task runs. Valid values are:

- Monday
- Tuesday
- Wednesday
- Thursday
- Friday
- Saturday
- Sunday

The acceptable values for this parameter are:

Friday	
Monday	
Saturday	
Sunday	
Thursday	
Tuesday	
Wednesday	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeleteThanOlderDays<Int32>

Specifies the number of days before Configuration Manager deletes the maintenance task.

Aliases	DeleteOlderThan
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeviceName<String>

Specifies the name of the device on which the maintenance task runs.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Enabled<Boolean>

Indicates whether the maintenance task is enabled in Configuration Manager.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnabledAlert<Boolean>

Indicates whether alerts for the maintenance task are enabled in Configuration Manager.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-LatestBeginTime<DateTime>

Specifies a future date and time at which the maintenance task runs.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-MaintenanceTask<MaintenanceTask>

Specifies the name of a maintenance task. Valid values are:

-- BackupConfigMgrSecondarySiteServer

- BackupSiteServer
- CheckApplicationTitleWithInventoryInformation
- ClearUndiscoveredClients
- DeleteAgedApplicationRequestData
- DeleteAgedClientAccessLicenseData
- DeleteAgedClientOperations
- DeleteAgedClientPresenceHistory
- DeleteAgedCollectedFiles
- DeleteAgedComputerAssociationData
- DeleteAgedConfigurationManagementData
- DeleteAgedDeleteDetectionData
- DeleteAgedDevicesManagedByTheExchangeServerConnector
- DeleteAgedDeviceWipeRecord
- DeleteAgedDiscoveryData
- DeleteAgedEndpointProtectionHealthStatusHistoryData
- DeleteAgedEnrolledDevices
- DeleteAgedInventoryHistory
- DeleteAgedLogData
- DeleteAgedNotificationTaskHistory
- DeleteAgedReplicationSummaryData
- DeleteAgedReplicationTrackingData
- DeleteAgedSoftwareMeteringData
- DeleteAgedStatusMessages
- DeleteAgedThreatData
- DeleteAgedUnknownComputers
- DeleteAgedUserDeviceAffinityData
- DeletedAgedClientPresenceHistory
- DeleteExpiredActivities
- DeleteExpiredActivityFacts
- DeleteExpiredBookmarks
- DeleteInactiveClientDiscoveryData
- DeleteObsoleteAlerts
- DeleteObsoleteClientDiscoveryData
- DeleteObsoleteForestDiscoverySitesAndSubnets
- DeleteUnusedApplicationRevisions
- EvaluateProvisionedAmtComputerCertificates
- ExportSiteDatabaseTransactionLog
- MonitorKeys
- RebuildIndexes

- ResetAmtComputerPasswords
- SiteDatabase
- SummarizeClientAccessLicenseWeeklyUsageData
- SummarizeInstalledSoftwareData
- SummarizeSoftwareMeteringFileUsageData
- SummarizeSoftwareMeteringMonthlyUsageData
- UpdateStatistics

The acceptable values for this parameter are:

BackupSiteServer	
CheckApplicationTitleWithInventoryInformation	
ClearUndiscoveredClients	
DeleteAgedApplicationRequestData	
DeleteAgedClientOperations	
DeleteAgedClientPresenceHistory	
DeleteAgedCollectedFiles	
DeleteAgedComputerAssociationData	
DeleteAgedDeleteDetectionData	
DeleteAgedDevicesManagedByTheExchangeServerConnector	
DeleteAgedDeviceWipeRecord	
DeleteAgedDiscoveryData	
DeleteAgedDistributionPointUsageStats	
DeleteAgedEndpointProtectionHealthStatusHistoryData	
DeleteAgedEnrolledDevices	
DeleteAgedInventoryHistory	
DeleteAgedLogData	
DeleteAgedNotificationTaskHistory	
DeleteAgedReplicationSummaryData	
DeleteAgedReplicationTrackingData	
DeleteAgedSoftwareMeteringData	
DeleteAgedSoftwareMeteringSummaryData	
DeleteAgedStatusMessages	
DeleteAgedThreatData	

DeleteAgedUnknownComputers	
DeleteAgedUserDeviceAffinityData	
DeleteInactiveClientDiscoveryData	
DeleteObsoleteAlerts	
DeleteObsoleteClientDiscoveryData	
DeleteObsoleteForestDiscoverySitesAndSubnets	
DeleteUnusedApplicationRevisions	
EvaluateProvisionedAmtComputerCertificates	
MonitorKeys	
RebuildIndexes	
SummarizeInstalledSoftwareData	
SummarizeSoftwareMeteringFileUsageData	
SummarizeSoftwareMeteringMonthlyUsageData	

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RunIntervalMinutes<Int32>

Specifies the number of minutes that elapse between each run for the maintenance task.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RunNow<Boolean>

Indicates whether Configuration Manager runs the maintenance task immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteCode<String>

Specifies the site code of the Configuration Manager site that hosts the site system role.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SummaryTask<SummaryTask>

Specifies a summary maintenance task. The acceptable value for this parameter is UpdateApplicationCatalogTables.

The acceptable values for this parameter are:

UpdateApplicationCatalogTables	
--------------------------------	--

Aliases	none
Required?	true

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Set a maintenance task to run once a week

This command specifies that the maintenance task named Backup runs on Friday each week on the Configuration Manager site that has the site code CM1.

```
PS C:\> Set-CMSiteMaintenanceTask -SiteCode "CM1" -MaintenanceTaskName "Backup" -DaysOfWeek  
Friday
```

Related topics

[Get-CMSiteMaintenanceTask](#)

Set-CMSiteSystemServer

Set-CMSiteSystemServer

Modifies an object that represents a site system server in System Center 2012 Configuration Manager.

Syntax

Parameter Set: SearchByNameMandatory

```
Set-CMSiteSystemServer -SiteCode <String> -SiteSystemServerName <String> [-AccountName <String> ] [-EnableProxy <Boolean> ] [-FdmOperation <Boolean> ] [-ProxyAccessAccount <IResultObject> ] [-ProxyServerName <String> ] [-ProxyServerPort <UInt32> ] [-PublicFqdn <String> ] [-UseSiteServerAccount] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Set-CMSiteSystemServer -InputObject <IResultObject> [-AccountName <String> ] [-EnableProxy <Boolean> ] [-FdmOperation <Boolean> ] [-ProxyAccessAccount <IResultObject> ] [-ProxyServerName <String> ] [-ProxyServerPort <UInt32> ] [-PublicFqdn <String> ] [-UseSiteServerAccount] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Set-CMSiteSystemServer** cmdlet modifies an object that represents a site system server in Microsoft System Center 2012 Configuration Manager. A site system server provides functionality to a configuration management site, such as communication between a System Center 2012 Configuration Manager server and System Center 2012 Configuration Manager clients.

Parameters

-AccountName<String>

Specifies an account name for the Configuration Manager site.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false

Accept Wildcard Characters?	false
-----------------------------	-------

-EnableProxy<Boolean>

Indicates whether to enable a proxy server.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

-FdmOperation<Boolean>

Indicates whether a site system pushes information back to a site server, or whether a site server pushes information to a site system.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies an input object. To obtain an input object, use the **Get-CMSiteSystemServer** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ProxyAccessAccount<IResultObject>

Specifies an input object that contains the domain and user name to authenticate with the proxy server. Do not use User Principal Name (UPN) format.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

-ProxyServerName<String>

Specifies the name of a proxy server.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

-ProxyServerPort<UInt32>

Specifies the port number of a proxy server.

Aliases	none
Required?	false
Position?	named

Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

-PublicFqdn<String>

Specifies a fully qualified domain name (FQDN) path for a Configuration Manager site.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

-SiteCode<String>

Specifies a site code in a Configuration Manager site.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

-SiteSystemServerName<String>

Specifies a server name in a Configuration Manager site.

Aliases	none
Required?	true
Position?	named

Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

-UseSiteServerAccount

Indicates that the cmdlet uses the site server account.

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

Example 1: Modify settings for a site system server

This command uses the **Set-CMSiteSystemServer** cmdlet to modify the settings for a site system server named Server1.Corp.Contoso.com. The command enables a proxy and specifies a proxy server and port number.

```
PS C:\> Set-CMSiteSystemServer -AccountName "Western\ElisaDaugherty" -EnableProxy $True -ProxyAddress "Proxy1.Corp.Contoso.Com" -ProxyPort 80 -SiteCode "CM1" -SiteSystemServerName "Server1.Corp.Contoso.Com" -UseSiteServerAccount
```

Related topics

[Get-CMSiteSystemServer](#)

[New-CMSiteSystemServer](#)

Set-CMSoftwareDistributionComponent

Set-CMSoftwareDistributionComponent

Sets properties of a software distribution component in Configuration Manager.

Syntax

Parameter Set: SearchBySiteCodeMandatory

```
Set-CMSoftwareDistributionComponent -SiteCode <String> [-DelayBeforeRetryingMinutes <Int32> ] [-MaximumPackageCount <Int32> ] [-MaximumThreadsPerPackage <Int32> ] [-MulticastDelayBeforeRetryingMinutes <Int32> ] [-MulticastRetryCount <Int32> ] [-NetworkAccessAccountName <String> ] [-RetryCount <Int32> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchBySiteCodeMandatory_ClientComputerAccount

```
Set-CMSoftwareDistributionComponent -SiteCode <String> [-ClientComputerAccount] [-DelayBeforeRetryingMinutes <Int32> ] [-MaximumPackageCount <Int32> ] [-MaximumThreadsPerPackage <Int32> ] [-MulticastDelayBeforeRetryingMinutes <Int32> ] [-MulticastRetryCount <Int32> ] [-RetryCount <Int32> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Set-CMSoftwareDistributionComponent** cmdlet sets properties of a software distribution component in Microsoft System Center 2012 Configuration Manager. You can configure the properties of an object to meet the demands that clients place on the System Center 2012 Configuration Manager site.

Parameters

-ClientComputerAccount

Indicates that the cmdlet uses a client computer account.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DelayBeforeRetryingMinutes<Int32>

Specifies a delay, in minutes, between software distribution attempts.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-MaximumPackageCount<Int32>

Specifies a maximum number of packages.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-MaximumThreadsPerPackage<Int32>

Specifies a maximum number of threads per package.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-MulticastDelayBeforeRetryingMinutes<Int32>

Specifies a delay, in minutes, between multicast software distribution attempts.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-MulticastRetryCount<Int32>

Specifies a retry count for multicast software distribution attempts.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-NetworkAccessAccountName<String>

Specifies an account name for network access.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RetryCount<Int32>

Specifies a retry count.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteCode<String>

Specifies a site code of a Configuration Manager site.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Set properties of a software distribution component

The following command sets all properties for a software distribution component.

```
PS C:\> Set-CMSoftwareDistributionComponent -SiteCode "CM2" -MaximumPackageCount 3 -
MaximumThreadsPerPackage 6 -RetryCount 99 -DelayBeforeRetryingMinutes 31 -
MulticastRetryCount 4 -MulticastDelayBeforeRetryingMinutes 2 -NetworkAccessAccount
"Western\ElisaDaugherty"
```

Related topics

[Get-CMSoftwareDistributionComponent](#)

Set-CMSoftwareInventory

Set-CMSoftwareInventory

Modifies an object that collects software inventory data on files.

Syntax

Parameter Set: SetById

```
Set-CMSoftwareInventory -Id <String[]> [-FamilyId <Int32> ] [-NewName <String> ] [-Publisher <String> ] [-Tag1Id <Int32> ] [-Tag2Id <Int32> ] [-Tag3Id <Int32> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetByName

```
Set-CMSoftwareInventory -Name <String[]> [-FamilyId <Int32> ] [-NewName <String> ] [-Publisher <String> ] [-Tag1Id <Int32> ] [-Tag2Id <Int32> ] [-Tag3Id <Int32> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetValueMandatory

```
Set-CMSoftwareInventory -InputObject <IResultObject> [-FamilyId <Int32> ] [-NewName <String> ] [-Publisher <String> ] [-Tag1Id <Int32> ] [-Tag2Id <Int32> ] [-Tag3Id <Int32> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Set-CMSoftwareInventory** cmdlet modifies an object that collects information about files on client devices in Microsoft System Center 2012 Configuration Manager.

Parameters

-FamilyId<Int32>

Specifies the ID of the family used to classify inventoried software in Configuration Manager.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-Id<String[]>

Specifies an array of IDs of software files.

Aliases	SoftwareKey
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a **CMSSoftwareInventory** object. To obtain a **CMSSoftwareInventory** object, use the **Get-
CMSSoftwareInventory** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String[]>

Specifies an array of names of software files.

Aliases	CommonName
Required?	true
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	true

-NewName<String>

Specifies a new name for inventoried software in Configuration Manager.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Publisher<String>

Specifies the name of a software publisher in Configuration Manager.

Aliases	CommonPublisher
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Tag1Id<Int32>

Specifies the ID of a tag to classify inventoried software in Configuration Manager.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Tag2Id<Int32>

Specifies the ID of a tag to classify inventoried software in Configuration Manager.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Tag3Id<Int32>

Specifies the ID of a tag to classify inventoried software in Configuration Manager.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Set a software inventory object

This command starts collecting software inventory data on the file named MSXML 6.0 Parser.

```
PS C:\> Set-CMSoftwareInventory -Name "MSXML 6.0 Parser"
```

Related topics

[Get-CMSoftwareInventory](#)

[Undo-CMSoftwareInventory](#)

Set-CMSoftwareMeteringRule

Set-CMSoftwareMeteringRule

Changes properties and security scopes for Configuration Manager software metering rules.

Syntax

Parameter Set: SetById

```
Set-CMSoftwareMeteringRule -Id <String[]> [-Comment <String> ] [-FileVersion <String> ] [-LanguageId <Int32> ] [-NewProductName <String> ] [-OriginalFileName <String> ] [-Path <String> ] [-SiteCode <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetByName

```
Set-CMSoftwareMeteringRule -ProductName <String> [-Comment <String> ] [-FileVersion <String> ] [-LanguageId <Int32> ] [-NewProductName <String> ] [-OriginalFileName <String> ] [-Path <String> ] [-SiteCode <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetValueMandatory

```
Set-CMSoftwareMeteringRule -InputObject <IResultObject> [-Comment <String> ] [-FileVersion <String> ] [-LanguageId <Int32> ] [-NewProductName <String> ] [-OriginalFileName <String> ] [-Path <String> ] [-SiteCode <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetSecurityScopeById

```
Set-CMSoftwareMeteringRule -Id <String[]> -SecurityScopeAction {AddMembership | RemoveMembership} -SecurityScopeName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetSecurityScopeByName

```
Set-CMSoftwareMeteringRule -ProductName <String> -SecurityScopeAction {AddMembership | RemoveMembership} -SecurityScopeName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetSecurityScopeByValue

```
Set-CMSoftwareMeteringRule -InputObject <IResultObject> -SecurityScopeAction {AddMembership | RemoveMembership} -SecurityScopeName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Set-CMSoftwareMeteringRule** cmdlet changes properties for software metering rules in Microsoft System Center 2012 Configuration Manager and adds or removes security scopes for software metering rules. Every rule must have at least one security scope.

Software metering monitors and collects software usage data from System Center 2012 Configuration Manager clients, such as when clients began using a particular software program and

how long users have worked with that software. You can create software metering rules that specify which software to monitor.

To change rule properties, you can specify rules to change by ID or by product name, or use the **Get-CMSoftwareMeteringRule** cmdlet. Likewise, you can change security scope for rules for specified ID, product name, or by using **Get-CMSoftwareMeteringRule**.

For more information about software metering in System Center 2012 Configuration Manager, see [Introduction to Software Metering in Configuration Manager](http://go.microsoft.com/fwlink/?LinkId=268432) (http://go.microsoft.com/fwlink/?LinkId=268432) on TechNet. For more information about security scopes, see [Planning for Security in Configuration Manager](http://go.microsoft.com/fwlink/?LinkId=268426) (http://go.microsoft.com/fwlink/?LinkId=268426).

Parameters

-Comment<String>

Specifies a comment for a software metering rule.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-FileVersion<String>

Specifies a version of the software that a rule meters.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Id<String[]>

Specifies an array of IDs for software metering rules.

Aliases	RuleId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a software metering rule object. To obtain a software metering rule object, use the **Get-SoftwareMeteringRule** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-LanguageId<Int32>

Specifies a LocaleID of the software that a rule meters. For more information and a list of locale identifiers, see the [Locale IDs Assigned by Microsoft](http://go.microsoft.com/fwlink/?LinkId=262651) topic at <http://go.microsoft.com/fwlink/?LinkId=262651>.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-NewProductName<String>

Specifies a new name for the software that a rule meters.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OriginalFileName<String>

Specifies an original file name of the software that a rule meters. This parameter can differ from the *FileName* parameter if a user changed the name.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Path<String>

Specifies a file path of the software that a rule meters.

Aliases	FileName
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-ProductName<String>

Specifies a name for a product that a rule meters.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SecurityScopeAction<SecurityScopeActionType>

Specifies an action for a security scope. Valid values are: AddMembership and RemoveMembership.

The acceptable values for this parameter are:

AddMembership	
RemoveMembership	

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SecurityScopeName<String>

Specifies the name of a security scope. A security scope name can be Default or the name of a custom security scope.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteCode<String>

Specifies a site code of a Configuration Manager site.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Change locale for rules for a product

This command sets the locale ID for rules that include the product name Accounting Package.

```
PS C:\> Set-CMSoftwareMeteringRule -ProductName "Accounting Package" -LanguageID 1036
```

Example 2: Add a security scope to rules for a product

This command adds the security scope called Scope05 to rules for the product name Accounting Package.

```
PS C:\> Set-CMSoftwareMeteringRule -ProductName "Accounting Package" -SecurityScopeAction AddMembership -SecurityScopeName "Scope05"
```

Related topics

[Disable-CMSoftwareMeteringRule](#)

[Enable-CMSoftwareMeteringRule](#)

[Get-CMSoftwareMeteringRule](#)

[New-CMSoftwareMeteringRule](#)

[Remove-CMSoftwareMeteringRule](#)

Set-CMSoftwareMeteringSetting

Set-CMSoftwareMeteringSetting

Configures Configuration Manager software metering properties.

Syntax

Parameter Set: SetByName

```
Set-CMSoftwareMeteringSetting [-AutoCreateDisabledRule <Boolean> ] [-AutoCreatePercentage <Int32> ] [-AutoCreateThreshold <Int32> ] [-DataRetentionDayCount <Int32> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Set-CMSoftwareMeteringSetting** cmdlet configures software metering properties for Microsoft System Center 2012 Configuration Manager.

Software metering can use software inventory information to create software metering rules. When you select this feature, System Center 2012 Configuration Manager identifies multiple computers that use the same software and creates a rule to track that usage in the future. You decide how long to keep software metering data that System Center 2012 Configuration Manager uses to create rules.

To prevent System Center 2012 Configuration Manager from creating too many rules, you can specify what percentage of computers use a piece of software before System Center 2012 Configuration Manager creates a rule. You can also set a rule threshold. If the number of software metering rules exceeds this threshold, System Center 2012 Configuration Manager stops automatically creating rules.

When System Center 2012 Configuration Manager creates a rule automatically, it creates that rule as disabled. A disabled rule does not collect information from clients. You can use the **Enable-CMSoftwareMeteringRule** cmdlet to enable a rule. You can use the **Remove-CMSoftwareMeteringRule** cmdlet to remove unwanted rules.

Parameters

-AutoCreateDisabledRule<Boolean>

Specifies whether Configuration Manager automatically creates software metering rules. If this value is \$True, Configuration Manager automatically creates software metering rules. If this value is \$False, it does not automatically create rules.

When Configuration Manager creates rules, it creates them as disabled.

You can use the values set by other parameters of this cmdlet to limit creation of rules.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AutoCreatePercentage<Int32>

Specifies a percentage of computers that use a piece of software for Configuration Manager to create a rule. Software metering calculates the number of computers as all the computers monitored for software metering by Configuration Manager, not just for a single site. Valid values are integers from 1 through 99

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AutoCreateThreshold<Int32>

Specifies a number of software metering rules as a threshold. When Configuration Manager exceeds this threshold, it stops automatically creating rules. The number of rules counted for this threshold includes all software metering rules, not only those that Configuration Manager creates automatically. Valid values are integers from 1 through 1000.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-DataRetentionDayCount<Int32>

Specifies a number of days that Configuration Manager keeps software metering data.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Disable automatic rule creation

This command disables automatic rule creation. System Center 2012 Configuration Manager does not automatically create software metering rules after you run this command.

```
PS C:\> Set-CMSoftwareMeteringSetting -AutoCreateDisabledRule $False
```

Example 2: Configure automatic rule creation

This command enables automatic rule creation and sets properties for it. This command sets the percentage of computers that use a piece of software to 50 percent, the rule threshold to 200, and the amount of time System Center 2012 Configuration Manager keeps software metering data to 30 days.

```
PS C:\> Set-CMSoftwareMeteringSetting -AutoCreateDisabledRule $True -AutoCreatePercentage 50  
-AutoCreateThreshold 200 -DataRetentionDayCount 30
```

Example 3: Change automatic rule creation percentage

This command changes the percentage of computers that use a piece of software to 20 percent.

```
PS C:\> Set-CMSoftwareMeteringSetting -AutoCreatePercentage 20
```

Related topics

[Get-CMSoftwareMeteringSetting](#)

[Enable-CMSoftwareMeteringRule](#)

[Remove-CMSoftwareMeteringRule](#)

Set-CMSoftwareUpdate

Set-CMSoftwareUpdate

Changes configuration settings for software updates.

Syntax

Parameter Set: SetById

```
Set-CMSoftwareUpdate -Id <String[]> [-CustomSeverity {Critical | Important | Low | Moderate | None} ] [-MaximumExecutionMinutes <Int32> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetByName

```
Set-CMSoftwareUpdate -Name <String[]> [-CustomSeverity {Critical | Important | Low | Moderate | None} ] [-MaximumExecutionMinutes <Int32> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetValue

```
Set-CMSoftwareUpdate -InputObject <IResultObject> [-CustomSeverity {Critical | Important | Low | Moderate | None} ] [-MaximumExecutionMinutes <Int32> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Set-CMSoftwareUpdate** cmdlet changes configuration settings for one or more software updates. You can use this cmdlet to set the severity and the maximum run time for an update. A software update is an update to Windows or other software that Microsoft System Center 2012 Configuration Manager applies to a collection of computers.

Parameters

-CustomSeverity<CustomSeverityType>

Specifies the severity for the software update. Valid values are:

- Critical
- Important
- Low
- Moderate
- None

The acceptable values for this parameter are:

Critical	
Important	
Low	
Moderate	
None	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Id<String[]>

Specifies an array of IDs of software updates.

Aliases	Ciid
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a **CMSoftwareUpdate** object. To obtain the **CMSoftwareUpdate** object, use the **Get-
CMSoftwareUpdate** cmdlet.

Aliases	none
Required?	true

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-MaximumExecutionMinutes<Int32>

Specifies the maximum amount of time, in minutes, that a software update has to complete installation on client computers.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String[]>

Specifies an array of names of software updates.

Aliases	LocalizedDisplayName
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Change settings of a software update

This command changes the severity and maximum run time settings for the software update named Cumulative Security Update for ActiveX Killbits for Windows 7 for x64-based Systems (KB2618451).

```
PS C:\> Set-CMSoftwareUpdate -Name "Cumulative Security Update for ActiveX Killbits for Windows 7 for x64-based Systems (KB2618451)" -MaximumExecutionMinutes 60 -CustomSeverity None
```

Related topics

[Get-CMSoftwareUpdate](#)

[Save-CMSoftwareUpdate](#)

[Sync-CMSoftwareUpdate](#)

Set-CMSoftwareUpdateAutoDeploymentRule

Set-CMSoftwareUpdateAutoDeploymentRule

Modifies Configuration Manager deployment rules for automatic software updates.

Syntax

Parameter Set: SearchByNameMandatory

```
Set-CMSoftwareUpdateAutoDeploymentRule -Name <String> [-AddToExistingSoftwareUpdateGroup <Boolean> ] [-AlertTime <Int32> ] [-AlertTimeUnit {Days | Hours | Months | Weeks} ] [-AllowRestart <Boolean> ] [-AllowSoftwareInstallationOutsideMaintenanceWindow <Boolean> ] [-AllowUseMeteredNetwork <Boolean> ] [-ArticleId <String[]> ] [-AvailableImmediately <Boolean> ] [-AvailableTime <Int32> ] [-AvailableTimeUnit {Days | Hours | Months | Weeks} ] [-BulletinId <String[]> ] [-CollectionName <String> ] [-CustomSeverity {Critical | Important | Low | Moderate | None} ] [-DateReleasedOrRevised {Last10months | Last11months | Last12hours | Last14days | Last16hours | Last1day | Last1hour | Last1month | Last1year | Last20hours | Last21days | Last28days | Last2days | Last2hours | Last2months | Last3days | Last3hours | Last3months | Last4days | Last4hours | Last4months | Last5days | Last5months | Last6days | Last6months | Last7days | Last7months | Last8hours | Last8months | Last9months} ] [-DeadlineImmediately <Boolean> ] [-DeadlineTime <Int32> ] [-DeadlineTimeUnit {Days | Hours | Months | Weeks} ] [-DeployWithoutLicense <Boolean> ] [-Description <String> ] [-DisableOperationManager <Boolean> ] [-DownloadFromMicrosoftUpdate <Boolean> ] [-EnabledAfterCreate <Boolean> ] [-Force] [-GenerateOperationManagerAlert <Boolean> ] [-GenerateSuccessAlert <Boolean> ] [-Language <String[]> ] [-LanguageSelection <String[]> ] [-MicrosoftAsVendor <Boolean> ] [-NewName <String> ] [-NoInstallOnRemote <Boolean> ] [-NoInstallOnUnprotected <Boolean> ] [-Product <String[]> ] [-Required <String[]> ] [-RunType {DoNotRunThisRuleAutomatically | RunTheRuleAfterAnySoftwareUpdatePointSynchronization | RunTheRuleOnSchedule} ] [-Schedule <IResultObject> ] [-SendWakeUpPacket <Boolean> ] [-Severity {Critical | Important | Low | Moderate | None} ] [-SuccessPercentage <Int32> ] [-Superseded <Boolean> ] [-SuppressRestartServer <Boolean> ] [-SuppressRestartWorkstation <Boolean> ] [-Title <String[]> ] [-UpdateClassification <String[]> ] [-UpdateDescription <String[]> ] [-UseBranchCache <Boolean> ] [-UserNotification {DisplayAll | DisplaySoftwareCenterOnly | HideAll} ] [-UseUtc <Boolean> ] [-VerboseLevel {AllMessages | OnlyErrorMessages | OnlySuccessAndErrorMessages} ] [-WriteFilterHandling <Boolean> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Set-CMSoftwareUpdateAutoDeploymentRule -Id <String[]> [-AddToExistingSoftwareUpdateGroup <Boolean> ] [-AlertTime <Int32> ] [-AlertTimeUnit {Days | Hours | Months | Weeks} ] [-AllowRestart <Boolean> ] [-AllowSoftwareInstallationOutsideMaintenanceWindow <Boolean> ] [-AllowUseMeteredNetwork <Boolean> ] [-ArticleId <String[]> ] [-AvailableImmediately <Boolean> ] [-AvailableTime <Int32> ] [-AvailableTimeUnit {Days | Hours | Months | Weeks} ] [-BulletinId <String[]> ] [-CollectionName <String> ] [-CustomSeverity {Critical | Important | Low | Moderate | None} ] [-DateReleasedOrRevised {Last10months | Last11months | Last12hours | Last14days | Last16hours | Last1day | Last1hour | Last1month | Last1year | Last20hours | Last21days | Last28days | Last2days | Last2hours | Last2months | Last3days | Last3hours | Last3months | Last4days | Last4hours | Last4months | Last5days | Last5months | Last6days |
```

```
Last6months | Last7days | Last7months | Last8hours | Last8months | Last9months} ] [-
DeadlineImmediately <Boolean> ] [-DeadlineTime <Int32> ] [-DeadlineTimeUnit {Days | Hours |
Months | Weeks} ] [-DeployWithoutLicense <Boolean> ] [-Description <String> ] [-
DisableOperationManager <Boolean> ] [-DownloadFromMicrosoftUpdate <Boolean> ] [-
EnabledAfterCreate <Boolean> ] [-Force] [-GenerateOperationManagerAlert <Boolean> ] [-
GenerateSuccessAlert <Boolean> ] [-Language <String[]> ] [-LanguageSelection <String[]> ] [-
MicrosoftAsVendor <Boolean> ] [-NewName <String> ] [-NoInstallOnRemote <Boolean> ] [-
NoInstallOnUnprotected <Boolean> ] [-Product <String[]> ] [-Required <String[]> ] [-RunType
{DoNotRunThisRuleAutomatically | RunTheRuleAfterAnySoftwareUpdatePointSynchronization |
RunTheRuleOnSchedule} ] [-Schedule <IResultObject> ] [-SendWakeUpPacket <Boolean> ] [-
Severity {Critical | Important | Low | Moderate | None} ] [-SuccessPercentage <Int32> ] [-
Superseded <Boolean> ] [-SuppressRestartServer <Boolean> ] [-SuppressRestartWorkstation
<Boolean> ] [-Title <String[]> ] [-UpdateClassification <String[]> ] [-UpdateDescription
<String[]> ] [-UseBranchCache <Boolean> ] [-UserNotification {DisplayAll |
DisplaySoftwareCenterOnly | HideAll} ] [-UseUtc <Boolean> ] [-VerboseLevel {AllMessages |
OnlyErrorMessage} ] [-WriteFilterHandling <Boolean> ] [-
Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Set-CMSoftwareUpdateAutoDeploymentRule -InputObject <IResultObject> [-
AddToExistingSoftwareUpdateGroup <Boolean> ] [-AlertTime <Int32> ] [-AlertTimeUnit {Days |
Hours | Months | Weeks} ] [-AllowRestart <Boolean> ] [-
AllowSoftwareInstallationOutsideMaintenanceWindow <Boolean> ] [-AllowUseMeteredNetwork
<Boolean> ] [-ArticleId <String[]> ] [-AvailableImmediately <Boolean> ] [-AvailableTime
<Int32> ] [-AvailableTimeUnit {Days | Hours | Months | Weeks} ] [-BulletinId <String[]> ] [-
CollectionName <String> ] [-CustomSeverity {Critical | Important | Low | Moderate | None} ]
[-DateReleasedOrRevised {Last10months | Last11months | Last12hours | Last14days |
Last16hours | Last1day | Last1hour | Last1month | Last1year | Last20hours | Last21days |
Last28days | Last2days | Last2hours | Last2months | Last3days | Last3hours | Last3months |
Last4days | Last4hours | Last4months | Last5days | Last5months | Last6days | Last6months |
Last7days | Last7months | Last8hours | Last8months | Last9months} ] [-DeadlineImmediately
<Boolean> ] [-DeadlineTime <Int32> ] [-DeadlineTimeUnit {Days | Hours | Months | Weeks} ] [-
DeployWithoutLicense <Boolean> ] [-Description <String> ] [-DisableOperationManager
<Boolean> ] [-DownloadFromMicrosoftUpdate <Boolean> ] [-EnabledAfterCreate <Boolean> ] [-
Force] [-GenerateOperationManagerAlert <Boolean> ] [-GenerateSuccessAlert <Boolean> ] [-
Language <String[]> ] [-LanguageSelection <String[]> ] [-MicrosoftAsVendor <Boolean> ] [-
NewName <String> ] [-NoInstallOnRemote <Boolean> ] [-NoInstallOnUnprotected <Boolean> ] [-
Product <String[]> ] [-Required <String[]> ] [-RunType {DoNotRunThisRuleAutomatically |
RunTheRuleAfterAnySoftwareUpdatePointSynchronization | RunTheRuleOnSchedule} ] [-Schedule
<IResultObject> ] [-SendWakeUpPacket <Boolean> ] [-Severity {Critical | Important | Low |
Moderate | None} ] [-SuccessPercentage <Int32> ] [-Superseded <Boolean> ] [-
SuppressRestartServer <Boolean> ] [-SuppressRestartWorkstation <Boolean> ] [-Title
<String[]> ] [-UpdateClassification <String[]> ] [-UpdateDescription <String[]> ] [-
UseBranchCache <Boolean> ] [-UserNotification {DisplayAll | DisplaySoftwareCenterOnly |
HideAll} ] [-UseUtc <Boolean> ] [-VerboseLevel {AllMessages | OnlyErrorMessage} ] [-WriteFilterHandling <Boolean> ] [-Confirm] [-WhatIf] [
<CommonParameters>]
```

Detailed Description

The **Set-CMSoftwareUpdateAutoDeploymentRule** cmdlet modifies Microsoft System Center 2012 Configuration Manager deployment rules for automatic software updates. To create a rule, use the **Get-CMSoftwareUpdateAutoDeploymentRule** cmdlet.

System Center 2012 Configuration Manager uses rules to manage automatic deployment of software updates. When a rule runs, System Center 2012 Configuration Manager adds updates that qualify for the rule to a software update group. The System Center 2012 Configuration Manager server downloads content files and copies them to distribution points, and then updates client computers.

Parameters

-AddToExistingSoftwareUpdateGroup<Boolean>

Indicates whether the rule adds to an existing update group. If this value is \$True, each time the rule runs and finds new updates, it adds them to an existing update group. If this value is \$False, it creates a new update group. Specify the existing update group or assign a name for the new update group by using the *DeploymentPackageName* parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AlertTime<Int32>

Specifies an integer offset from an update deployment deadline. The rule uses this value to specify when the rule generates alerts. Specify a time unit by using the *AlertTimeUnit* parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AlertTimeUnit<TimeUnitType>

Specifies a unit of time for the *AlertTime* parameter. Valid values are:

- Days
- Hours
- Months
- Weeks

The acceptable values for this parameter are:

Days	
Hours	
Months	
Weeks	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AllowRestart<Boolean>

Indicates whether to allow a computer to restart if the update deployment takes place outside of a maintenance window. A maintenance window is a specified period of time used for computer maintenance and updates. If this value is \$True, this Configuration Manager restarts the computer, if necessary to complete the update. If this value is \$False, Configuration Manager does not restart the computer.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-AllowSoftwareInstallationOutsideMaintenanceWindow<Boolean>

Indicates whether the update deployment takes place even if scheduled outside of a maintenance window. A maintenance window is a specified period of time used for computer maintenance and updates. If this value is \$True, this Configuration Manager deploys the update even the scheduled time falls outside the service window. If this value is \$False, Configuration Manager does not deploy the update outside the service window, but Configuration Manager waits until it can deploy in a service window.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AllowUseMeteredNetwork<Boolean>

Indicates whether to allow clients to download content over a metered Internet connection after the deadline, which may incur additional expense.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ArticleId<String[]>

Specifies an array of criteria, as strings, for software updates. The rule adds software updates that have article IDs that meet specified criteria to the software update group.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AvailableImmediately<Boolean>

Indicates whether this rule deploys updates as soon as the updates become available. If you select a value of `$False`, use the *AvailableTime* and *AvailableTimeUnit* parameters to specify how long after the rule runs to deploy the updates.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AvailableTime<Int32>

Specifies a period of time as an integer. Configuration Manager deploys the updates this long after the rule runs. Specify a time unit by using the *AvailableTimeUnit* parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AvailableTimeUnit<TimeUnitType>

Specifies a unit of time for the *AvailableTime* parameter. Valid values are:

- Days
- Hours
- Months
- Weeks

The acceptable values for this parameter are:

Days	
Hours	
Months	
Weeks	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-BulletinId<String[]>

Specifies an array of criteria, as strings, for software updates. The rule adds software updates that have bulletin IDs that meet specified criteria to the software update group.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CollectionName<String>

Specifies the name of device collection or user collection.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CustomSeverity<SeverityType[]>

Specifies an array of custom severity types for software updates. The rule adds software updates that have custom severity levels that meet specified criteria to the software update group. Valid values are:

- Critical
- Important
- Low
- Moderate
- None

The acceptable values for this parameter are:

Critical	
Important	
Low	
Moderate	
None	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DateReleasedOrRevised<DateReleasedOrRevisedType>

Specifies a date released or revised for software updates. The rule adds software updates that have a date that meets specified criteria to the software update group. Valid values are:

- Last10months
- Last11months
- Last12hours
- Last14days
- Last16hours
- Last1day
- Last1hour
- Last1month
- Last1year
- Last20hours
- Last21days
- Last28days
- Last2days
- Last2hours
- Last2months
- Last3days
- Last3hours
- Last3months
- Last4days
- Last4hours
- Last4months
- Last5days
- Last5months
- Last6days
- Last6months
- Last7days
- Last7months
- Last8hours
- Last8months
- Last9months

The acceptable values for this parameter are:

Last10months	
--------------	--

Last11months	
Last12hours	
Last14days	
Last16hours	
Last1day	
Last1hour	
Last1month	
Last1year	
Last20hours	
Last21days	
Last28days	
Last2days	
Last2hours	
Last2months	
Last3days	
Last3hours	
Last3months	
Last4days	
Last4hours	
Last4months	
Last5days	
Last5months	
Last6days	
Last6months	
Last7days	
Last7months	
Last8hours	
Last8months	
Last9months	

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeadlineImmediately<Boolean>

Indicates whether to impose the deadline as soon as the rule runs. If you specify a value of `$False`, use the *DeadlineTime* and *DeadlineTimeUnit* parameters to specify how long after the rule runs to set the deadline. After the deadline, Configuration Manager installs required updates.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeadlineTime<Int32>

Specifies a period of time as an integer. The deadline for updates is this long after the rule runs. Specify a time unit by using the *DeadlineTimeUnit* parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeadlineTimeUnit<TimeUnitType>

Specifies a unit of time for the *DeadlineTime* parameter. Valid values are:

- Days
- Hours
- Months
- Weeks

The acceptable values for this parameter are:

Days	
Hours	
Months	
Weeks	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeployWithoutLicense<Boolean>

Indicates whether the rule deploys updates without licenses. If you specify a value of \$True, Configuration Manager deploys all updates for this rule and approves any license agreements. If this value is \$False, Configuration Manager deploys only updates that do not include a license or for which the license agreement has been approved.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Description<String>

Specifies a description for the automatic deployment rule for software updates.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DisableOperationManager<Boolean>

Indicates whether to disable System Center 2012 – Operations Manager alerts during software updates.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DownloadFromMicrosoftUpdate<Boolean>

Indicates whether computers download content from Microsoft Update if that content is unavailable on a preferred distribution point of remote distribution point.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnabledAfterCreate<Boolean>

Indicates whether to enable software deployment for the associated software update group after this rule runs. If this value is \$False, deploy the software update group manually.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-GenerateOperationManagerAlert<Boolean>

Indicates whether to generate Operations Manager alerts during a software update.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-GenerateSuccessAlert<Boolean>

Indicates whether to generate an alert for successful deployment.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Id<String[]>

Specifies an array of IDs of automatic deployment rules for software updates.

Aliases	AutoDeploymentId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies an object representing an automatic deployment rule for software updates. To obtain an object representing a rule, use the **Get-CMSoftwareUpdateAutoDeploymentRule** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Language<String[]>

Specifies an array of criteria, as strings, for software updates. The rule adds software updates that have languages that meet specified criteria to the software update group.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-LanguageSelection<String[]>

Specifies an array of languages, as strings. Computers download software updates available in the specified languages, in addition to non-language-specific updates.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-MicrosoftAsVendor<Boolean>

Indicates whether the rule includes only updates that have Microsoft as the vendor.

Aliases	none
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies a name for the automatic deployment rule for software updates.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-NewName<String>

Specifies a new name for the automatic deployment rule for software updates.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-NoInstallOnRemote<Boolean>

Indicates whether to disallow installation of updates on remote systems. If you specify a value of \$True, if the client is within a slow or unreliable network boundary, or when the client uses a fallback source location for content, then Configuration Manager does not install software updates. If you specify a value of \$False, installation proceeds.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-NoInstallOnUnprotected<Boolean>

Indicates whether to disallow installation of updates on unprotected systems. If you specify a value of \$True, if software updates are not available on any preferred distribution points, Configuration Manager does not download and install software updates. If you specify a value of \$False, installation proceeds.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Product<String[]>

Specifies an array of criteria, as strings, for software updates. The rule adds software updates for products that meet specified criteria to the software update group.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Required<String[]>

Specifies an array of criteria, as strings, for software updates. The rule adds software updates identified by required that meet specified criteria to the software update group.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RunType<RunType>

Specifies the mode in which an update runs on the client computer. Valid values are:

- DoNotRunThisRuleAutomatically
- RunTheRuleAfterAnySoftwareUpdatePointSynchronization
- RunTheRuleOnSchedule

If you specify RunTheRuleOnSchedule, specify a schedule by using the *Schedule* parameter.

The acceptable values for this parameter are:

DoNotRunThisRuleAutomatically	
RunTheRuleAfterAnySoftwareUpdatePointSynchronization	
RunTheRuleOnSchedule	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Schedule<IResultObject>

Specifies a schedule object for the deployment. To obtain a schedule object, use the **New-CMSchedule** cmdlet. Specify a schedule for this parameter if you specify a value of *RunTheRuleOnSchedule* for the *RunType* parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SendWakeUpPacket<Boolean>

Indicates whether to send a wake up packet to computers before the deployment begins. If this value is *\$True*, Configuration Manager wakes a computer from sleep. If this value is *\$False*, it does not wake computers from sleep. For computers to wake, you must first configure Wake On LAN.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Severity<SeverityType[]>

Specifies an array of severity levels for software updates. The rule adds software updates for specified severity types to the software update group. Valid values are:

- Critical
- Important
- Low
- Moderate
- None

The acceptable values for this parameter are:

Critical	
Important	
Low	
Moderate	
None	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SuccessPercentage<Int32>

Specifies a percentage for client compliance as an integer from 0 to 99. If compliance falls below this percentage, Configuration Manager produces optional alerts.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Superseded<Boolean>

Indicates whether the rule adds updates superseded by other updates.

Aliases	none
Required?	false
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SuppressRestartServer<Boolean>

Indicates whether to suppress a required update for a server. Some software updates require a system restart to complete the installation process.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SuppressRestartWorkstation<Boolean>

Indicates whether to suppress a required update for a workstation. Some software updates require a system restart to complete the installation process.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Title<String[]>

Specifies an array of criteria, as strings, for software updates. The rule adds software updates that have titles that meet specified criteria to the software update group.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UpdateClassification<String[]>

Specifies an array of criteria, as strings, for software updates. The rule adds software updates that have update classifications that meet specified criteria to the software update group.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UpdateDescription<String[]>

Specifies an array of criteria, as strings, for software updates. The rule adds software updates that have update descriptions that meet specified criteria to the software update group.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UseBranchCache<Boolean>

Indicates whether to use a branch cache for this update deployment. If you specify a value of \$True, clients share content on the same subnet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UserNotification<UserNotificationType>

Specifies the type of user notification. Valid values are:

- DisplayAll. Display in Software Center and show all notifications.
- DisplaySoftwareCenterOnly. Display in Software Center, and only show notifications of computer restarts.
- HideAll. Hide in Software Center and all notifications.

The acceptable values for this parameter are:

DisplayAll	
DisplaySoftwareCenterOnly	
HideAll	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UseUtc<Boolean>

Indicates whether to use Coordinated Universal Time (UTC), also known as Greenwich Mean Time. If this value is \$True, Configuration Manager uses UTC for this deployment. If this value is \$False, Configuration Manager uses local time.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-VerboseLevel<VerboseLevelType>

Specifies the level of detail you want clients to report for deployments that this rule creates. Valid values are:

- AllMessages
- OnlyErrorMessages
- OnlySuccessAndErrorMessages

The acceptable values for this parameter are:

AllMessages	
OnlyErrorMessages	
OnlySuccessAndErrorMessages	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-WriteFilterHandling<Boolean>

Indicates whether to enable write filters for embedded devices. For a value of \$True, the device commits changes during a maintenance window. This action requires a restart. For a value of \$False, the device saves changes in an overlay and commits them later.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Modify the language selection and name of a rule

This command modifies the automatic deployment rule named DeploymentRule07. The command specifies Portuguese (Brazil) as the language selection. The command also changes the name of the rule to DeploymentRule07Revised.

```
PS C:\> Set-CMSoftwareUpdateAutoDeploymentRule -Name "DeploymentRule07" -NewName  
"DeploymentRule07Revised" -Description "Update rule to use Portuguese (Brazil)." -  
LanguageSelection "Portuguese (Brazil)"
```

Related topics

[Disable-CMSoftwareUpdateAutoDeploymentRule](#)

[Enable-CMSoftwareUpdateAutoDeploymentRule](#)

[Get-CMSoftwareUpdateAutoDeploymentRule](#)

[Invoke-CMSoftwareUpdateAutoDeploymentRule](#)

[New-CMSoftwareUpdateAutoDeploymentRule](#)

[Remove-CMSoftwareUpdateAutoDeploymentRule](#)

Set-CMSoftwareUpdateBasedClientInstallation

Set-CMSoftwareUpdateBasedClientInstallation

Modifies a client installation on a Configuration Manager software update point.

Syntax

Parameter Set: SearchBySiteCodeMandatory

```
Set-CMSoftwareUpdateBasedClientInstallation -EnableWsus <Boolean> -SiteCode <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByNameMandatory

```
Set-CMSoftwareUpdateBasedClientInstallation -EnableWsus <Boolean> -Name <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Set-CMSoftwareUpdateBasedClientInstallation** cmdlet modifies a client installation hosted on a software update point for Microsoft System Center 2012 Configuration Manager.

Parameters

-EnableWsus<Boolean>

Indicates whether to enable Windows Server Update Service (WSUS).

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies a name in Configuration Manager.

Aliases	SiteName
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteCode<String>

Specifies the Configuration Manager site for which the client installation method is configured.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Modify a client installation to enable WSUS

This command enables WSUS for a client installation.

```
PS C:\> Set-CMSoftwareUpdateBasedClientInstallation -EnableWSUS $True -SiteCode "CM1"
```

Related topics

[Get-CMSoftwareUpdateBasedClientInstallation](#)

Set-CMSoftwareUpdateDeployment

Set-CMSoftwareUpdateDeployment

Modifies a software update deployment in Configuration Manager.

Syntax

Parameter Set: SetSoftwareUpdateDeploymentByIdMandatory

```
Set-CMSoftwareUpdateDeployment -DeploymentName <String> -SoftwareUpdateId <String> [-AlertDay <DateTime> ] [-AlertTime <DateTime> ] [-AllowRestart <Boolean> ] [-AllowUseMeteredNetwork <Boolean> ] [-CollectionName <String> ] [-DeploymentAvailableDay <DateTime> ] [-DeploymentAvailableTime <DateTime> ] [-DeploymentExpireDay <DateTime> ] [-DeploymentExpireTime <DateTime> ] [-DeploymentType {Available | Required} ] [-Description <String> ] [-DisableOperationsManagerAlert <Boolean> ] [-DownloadFromMicrosoftUpdate <Boolean> ] [-GenerateOperationsManagerAlert <Boolean> ] [-GenerateSuccessAlert <Boolean> ] [-NewDeploymentName <String> ] [-PercentSuccess <Int32> ] [-PersistOnWriteFilterDevice <Boolean> ] [-ProtectedType {NoInstall | RemoteDistributionPoint} ] [-RestartServer <Boolean> ] [-RestartWorkstation <Boolean> ] [-SendWakeUpPacket <Boolean> ] [-SoftwareInstallation <Boolean> ] [-TimeBasedOn {LocalTime | UTC} ] [-UnprotectedType {NoInstall | UnprotectedDistributionPoint} ] [-UseBranchCache <Boolean> ] [-UserNotification {DisplayAll | DisplaySoftwareCenterOnly | HideAll} ] [-VerbosityLevel {AllMessages | OnlyErrorMessages | OnlySuccessAndErrorMessages} ] [-Confirm] [-WhatIf] [ <CommonParameters> ]
```

Parameter Set: SetSoftwareUpdateDeploymentByNameMandatory

```
Set-CMSoftwareUpdateDeployment -DeploymentName <String> -SoftwareUpdateName <String> [-AlertDay <DateTime> ] [-AlertTime <DateTime> ] [-AllowRestart <Boolean> ] [-AllowUseMeteredNetwork <Boolean> ] [-CollectionName <String> ] [-DeploymentAvailableDay <DateTime> ] [-DeploymentAvailableTime <DateTime> ] [-DeploymentExpireDay <DateTime> ] [-DeploymentExpireTime <DateTime> ] [-DeploymentType {Available | Required} ] [-Description <String> ] [-DisableOperationsManagerAlert <Boolean> ] [-DownloadFromMicrosoftUpdate <Boolean> ] [-GenerateOperationsManagerAlert <Boolean> ] [-GenerateSuccessAlert <Boolean> ] [-NewDeploymentName <String> ] [-PercentSuccess <Int32> ] [-PersistOnWriteFilterDevice <Boolean> ] [-ProtectedType {NoInstall | RemoteDistributionPoint} ] [-RestartServer <Boolean> ] [-RestartWorkstation <Boolean> ] [-SendWakeUpPacket <Boolean> ] [-SoftwareInstallation <Boolean> ] [-TimeBasedOn {LocalTime | UTC} ] [-UnprotectedType {NoInstall | UnprotectedDistributionPoint} ] [-UseBranchCache <Boolean> ] [-UserNotification {DisplayAll | DisplaySoftwareCenterOnly | HideAll} ] [-VerbosityLevel {AllMessages | OnlyErrorMessages | OnlySuccessAndErrorMessages} ] [-Confirm] [-WhatIf] [ <CommonParameters> ]
```

Parameter Set: SetSoftwareUpdateDeploymentByValueMandatory

```
Set-CMSoftwareUpdateDeployment -DeploymentName <String> -SoftwareUpdate <IResultObject> [-AlertDay <DateTime> ] [-AlertTime <DateTime> ] [-AllowRestart <Boolean> ] [-AllowUseMeteredNetwork <Boolean> ] [-CollectionName <String> ] [-DeploymentAvailableDay <DateTime> ] [-DeploymentAvailableTime <DateTime> ] [-DeploymentExpireDay <DateTime> ] [-DeploymentExpireTime <DateTime> ] [-DeploymentType {Available | Required} ] [-Description
```

<String>] [-DisableOperationsManagerAlert <Boolean>] [-DownloadFromMicrosoftUpdate <Boolean>] [-GenerateOperationsManagerAlert <Boolean>] [-GenerateSuccessAlert <Boolean>] [-NewDeploymentName <String>] [-PercentSuccess <Int32>] [-PersistOnWriteFilterDevice <Boolean>] [-ProtectedType {NoInstall | RemoteDistributionPoint}] [-RestartServer <Boolean>] [-RestartWorkstation <Boolean>] [-SendWakeUpPacket <Boolean>] [-SoftwareInstallation <Boolean>] [-TimeBasedOn {LocalTime | UTC}] [-UnprotectedType {NoInstall | UnprotectedDistributionPoint}] [-UseBranchCache <Boolean>] [-UserNotification {DisplayAll | DisplaySoftwareCenterOnly | HideAll}] [-VerbosityLevel {AllMessages | OnlyErrorMessages | OnlySuccessAndErrorMessages}] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SetSoftwareUpdateGroupDeploymentByIdMandatory

Set-CMSoftwareUpdateDeployment -DeploymentName <String> -SoftwareUpdateGroupId <String> [-AlertDay <DateTime>] [-AlertTime <DateTime>] [-AllowRestart <Boolean>] [-AllowUseMeteredNetwork <Boolean>] [-CollectionName <String>] [-DeploymentAvailableDay <DateTime>] [-DeploymentAvailableTime <DateTime>] [-DeploymentExpireDay <DateTime>] [-DeploymentExpireTime <DateTime>] [-DeploymentType {Available | Required}] [-Description <String>] [-DisableOperationsManagerAlert <Boolean>] [-DownloadFromMicrosoftUpdate <Boolean>] [-GenerateOperationsManagerAlert <Boolean>] [-GenerateSuccessAlert <Boolean>] [-NewDeploymentName <String>] [-PercentSuccess <Int32>] [-PersistOnWriteFilterDevice <Boolean>] [-ProtectedType {NoInstall | RemoteDistributionPoint}] [-RestartServer <Boolean>] [-RestartWorkstation <Boolean>] [-SendWakeUpPacket <Boolean>] [-SoftwareInstallation <Boolean>] [-TimeBasedOn {LocalTime | UTC}] [-UnprotectedType {NoInstall | UnprotectedDistributionPoint}] [-UseBranchCache <Boolean>] [-UserNotification {DisplayAll | DisplaySoftwareCenterOnly | HideAll}] [-VerbosityLevel {AllMessages | OnlyErrorMessages | OnlySuccessAndErrorMessages}] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SetSoftwareUpdateGroupDeploymentByNameMandatory

Set-CMSoftwareUpdateDeployment -DeploymentName <String> -SoftwareUpdateGroupName <String> [-AlertDay <DateTime>] [-AlertTime <DateTime>] [-AllowRestart <Boolean>] [-AllowUseMeteredNetwork <Boolean>] [-CollectionName <String>] [-DeploymentAvailableDay <DateTime>] [-DeploymentAvailableTime <DateTime>] [-DeploymentExpireDay <DateTime>] [-DeploymentExpireTime <DateTime>] [-DeploymentType {Available | Required}] [-Description <String>] [-DisableOperationsManagerAlert <Boolean>] [-DownloadFromMicrosoftUpdate <Boolean>] [-GenerateOperationsManagerAlert <Boolean>] [-GenerateSuccessAlert <Boolean>] [-NewDeploymentName <String>] [-PercentSuccess <Int32>] [-PersistOnWriteFilterDevice <Boolean>] [-ProtectedType {NoInstall | RemoteDistributionPoint}] [-RestartServer <Boolean>] [-RestartWorkstation <Boolean>] [-SendWakeUpPacket <Boolean>] [-SoftwareInstallation <Boolean>] [-TimeBasedOn {LocalTime | UTC}] [-UnprotectedType {NoInstall | UnprotectedDistributionPoint}] [-UseBranchCache <Boolean>] [-UserNotification {DisplayAll | DisplaySoftwareCenterOnly | HideAll}] [-VerbosityLevel {AllMessages | OnlyErrorMessages | OnlySuccessAndErrorMessages}] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SetSoftwareUpdateGroupDeploymentByValueMandatory

Set-CMSoftwareUpdateDeployment -DeploymentName <String> -SoftwareUpdateGroup <IResultObject> [-AlertDay <DateTime>] [-AlertTime <DateTime>] [-AllowRestart <Boolean>] [-AllowUseMeteredNetwork <Boolean>] [-CollectionName <String>] [-DeploymentAvailableDay <DateTime>] [-DeploymentAvailableTime <DateTime>] [-DeploymentExpireDay <DateTime>] [-DeploymentExpireTime <DateTime>] [-DeploymentType {Available | Required}] [-Description <String>] [-DisableOperationsManagerAlert <Boolean>] [-DownloadFromMicrosoftUpdate

```
<Boolean> ] [-GenerateOperationsManagerAlert <Boolean> ] [-GenerateSuccessAlert <Boolean> ]
[-NewDeploymentName <String> ] [-PercentSuccess <Int32> ] [-PersistOnWriteFilterDevice
<Boolean> ] [-ProtectedType {NoInstall | RemoteDistributionPoint} ] [-RestartServer
<Boolean> ] [-RestartWorkstation <Boolean> ] [-SendWakeUpPacket <Boolean> ] [-
SoftwareInstallation <Boolean> ] [-TimeBasedOn {LocalTime | UTC} ] [-UnprotectedType
{NoInstall | UnprotectedDistributionPoint} ] [-UseBranchCache <Boolean> ] [-UserNotification
{DisplayAll | DisplaySoftwareCenterOnly | HideAll} ] [-VerbosityLevel {AllMessages |
OnlyErrorMessages | OnlySuccessAndErrorMessages} ] [-Confirm] [-WhatIf] [
<CommonParameters>]
```

Detailed Description

The **Set-CMSoftwareUpdateDeployment** cmdlet modifies a deployment of software updates in Microsoft System Center 2012 Configuration Manager.

Parameters

-AlertDay<DateTime>

Specifies a day, in MM/DD/YYYY format, to notify clients of new software updates.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AlertTime<DateTime>

Specifies a time, in HH:MM format, to notify clients of new software updates.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-AllowRestart<Boolean>

Indicates whether to allow a restart following installation.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AllowUseMeteredNetwork<Boolean>

Indicates whether to allow clients to use a metered network to download updates.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CollectionName<String>

Specifies a name of a collection in Configuration Manager. A collection is a group of client computers.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-DeploymentAvailableDay<DateTime>

Specifies a day, in MM/DD/YYYY format, when a software update becomes available to clients. By default, the update is available immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeploymentAvailableTime<DateTime>

Specifies a time, in HH:MM format, when a software update becomes available to clients. By default, the update is available immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeploymentExpireDay<DateTime>

Specifies a day, in MM/DD/YYYY format, when a software update expires. To expire a software update on a certain day, set this parameter.

Aliases	none
Required?	false
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeploymentExpireTime<DateTime>

Specifies a time, in HH:MM format, when a software update expires. To expire a software update at a certain time, set this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeploymentName<String>

Specifies a name for a software update deployment in Configuration Manager.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeploymentType<DeploymentType>

Specifies a deployment type in Configuration Manager.

The acceptable values for this parameter are:

Available	
-----------	--

Required	
----------	--

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Description<String>

Specifies a description for a software update deployment.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DisableOperationsManagerAlert<Boolean>

Indicates whether to disable System Center 2012 – Operations Manager alerts during software updates.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DownloadFromMicrosoftUpdate<Boolean>

Indicates whether clients download updates directly from Microsoft Update.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-GenerateOperationsManagerAlert<Boolean>

Indicates whether to generate Operations Manager alerts when a software installation fails.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-GenerateSuccessAlert<Boolean>

Indicates whether to generate alerts when a software installation succeeds.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-NewDeploymentName<String>

Specifies a name for a new deployment in Configuration Manager.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PercentSuccess<Int32>

Specifies a percentage of the update.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PersistOnWriteFilterDevice<Boolean>

Indicates whether to install a software update on the temporary overlay and commit changes later, or commit the changes at an installation deadline or a maintenance window.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ProtectedType<ProtectedType>

Specifies a protected type.

The acceptable values for this parameter are:

NoInstall	
RemoteDistributionPoint	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RestartServer<Boolean>

Indicates whether to allow a server to restart following a software update.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RestartWorkstation<Boolean>

Indicates whether to allow a workstation to restart following a software update.

Aliases	none
Required?	false
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SendWakeUpPacket<Boolean>

Indicates whether to send a wake up packet to computers before the deployment begins. If this value is \$True, Configuration Manager wakes a computer from sleep. If this value is \$False, it does not wake computers from sleep. For computers to wake, you must first configure Wake On LAN.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SoftwareInstallation<Boolean>

Indicates whether to allow the software update to install, even if the installation occurs outside of a maintenance window.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SoftwareUpdate<IResultObject>

Specifies a software update deployment object. To obtain a software update deployment object, use the [Get-CMSoftwareUpdate](#) cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SoftwareUpdateGroup<IResultObject>

Specifies a software update group object. To obtain a software update group object, use the [Get-CMSoftwareUpdateGroup](#) cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SoftwareUpdateGroupId<String>

Specifies an ID for a software update group. A software update group contains individual software updates.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SoftwareUpdateGroupName<String>

Specifies a name for a software update group.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SoftwareUpdateId<String>

Specifies an ID for a software update in Configuration Manager.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SoftwareUpdateName<String>

Specifies a name for a software update in Configuration Manager.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-TimeBasedOn<TimeType>

Specifies that client computers use either local or UTC time to determine the availability of a program. UTC time makes the software update available at the same time for all computers.

The acceptable values for this parameter are:

LocalTime	
UTC	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UnprotectedType<UnprotectedType>

Specifies an unprotected type.

The acceptable values for this parameter are:

NoInstall	
UnprotectedDistributionPoint	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UseBranchCache<Boolean>

Indicates whether to use Branch Cache as a distribution point for updates.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UserNotification<UserNotificationType>

Specifies a user notification type.

The acceptable values for this parameter are:

DisplayAll	
DisplaySoftwareCenterOnly	
HideAll	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-VerbosityLevel<VerbosityLevelType>

Specifies a verbosity level type, such as error messages. Valid values are:

- AllMessages
- OnlyErrorMessages
- OnlySuccessandErrorMessages

The acceptable values for this parameter are:

AllMessages	
OnlyErrorMessage	
OnlySuccessAndErrorMessage	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Set a deployment with expiration time

This command sets a software update deployment by using a software update name and expiration time.

```
PS C:\> Set-CMSoftwareUpdateDeployment -SoftwareUpdateName "CT" -DeploymentName "Contoso-test1" -NewDeploymentName "Contoso-test5" -Description "Contoso-test5-deployment" -CollectionName "All Mobile Devices" -SendWakeUpPacket $False -VerbosityLevel OnlySuccessAndErrorMessage -TimeBasedOn LocalTime -DeploymentAvailableDay 2013/11/24 -DeploymentAvailableTime 13:26 -DeploymentExpireDay 2014/7/22 -DeploymentExpireTime 4:30 -UserNotification DisplayAll -SoftwareInstallation $False -AllowRestart $False -RestartServer $False -RestartWorkstation $False -PersistOnWriteFilterDevice $True -GenerateSuccessAlert $False -PercentSuccess 99 -DisableOperationsManagerAlert $False -GenerateOperationsManagerAlert $False -ProtectedType NoInstall -UnprotectedType UnprotectedDistributionPoint -UseBranchCache $True -DownloadFromMicrosoftUpdate $False -AllowUseMeteredNetwork $False
```

Example 2: Start a deployment without expiration time

This command sets a software update deployment by using a software update name but no specified expiration time.

```
PS C:\> Set-CMSoftwareUpdateDeployment -SoftwareUpdateName "CT" -DeploymentName "Contoso-test2" -NewDeploymentName "Contoso-test6" -Description "Contoso-test6-deployment" -CollectionName "All Mobile Devices" -VerbosityLevel OnlyErrorMessage -TimeBasedOn LocalTime -DeploymentAvailableDay 2013/12/24 -DeploymentAvailableTime 3:56 -UserNotification DisplaySoftwareCenterOnly -PersistOnWriteFilterDevice $True -DisableOperationsManagerAlert $False -GenerateOperationsManagerAlert $False -ProtectedType NoInstall -UnprotectedType UnprotectedDistributionPoint -UseBranchCache $True -DownloadFromMicrosoftUpdate $False -AllowUseMeteredNetwork $False
```

Example 3: Start a deployment by software update group name and expiration time

This command sets a software update deployment by using a software update group name and an expiration time.

```
PS C:\> Set-CMSoftwareUpdateDeployment -SoftwareUpdateGroupName "CTG" -DeploymentName "Contoso-test3" -NewDeploymentName "Contoso-test7" -Description "Contoso-test7-deployment" -CollectionName "All Mobile Devices" -SendWakeUpPacket $False -VerbosityLevel OnlySuccessAndErrorMessage -TimeBasedOn LocalTime -DeploymentAvailableDay 2013/11/24 -DeploymentAvailableTime 13:26 -DeploymentExpireDay 2014/7/22 -DeploymentExpireTime 4:30 -
```

```
UserNotification DisplayAll -SoftwareInstallation $False -AllowRestart $False -RestartServer $False -RestartWorkstation $False -PersistOnWriteFilterDevice $True -GenerateSuccessAlert $False -PercentSuccess 99 -DisableOperationsManagerAlert $False -GenerateOperationsManagerAlert $False -ProtectedType NoInstall -UnprotectedType UnprotectedDistributionPoint -UseBranchCache $True -DownloadFromMicrosoftUpdate $False -AllowUseMeteredNetwork $False
```

Example 4: Start a deployment by software update group name

This command starts a software update deployment by using a software update group name but no specified expiration time.

```
PS C:\> Set-CMSoftwareUpdateDeployment -SoftwareUpdateGroupName "CTG" -DeploymentName "Contoso-test4" -NewDeploymentName "Contoso-test8" -Description "Contoso-test8-deployment" -CollectionName "All Mobile Devices" -VerbosityLevel OnlyErrorMessages -TimeBasedOn LocalTime -DeploymentAvailableDay 2013/12/24 -DeploymentAvailableTime 3:56 -UserNotification DisplaySoftwareCenterOnly -PersistOnWriteFilterDevice $True -DisableOperationsManagerAlert $False -GenerateOperationsManagerAlert $False -ProtectedType NoInstall -UnprotectedType UnprotectedDistributionPoint -UseBranchCache $True -DownloadFromMicrosoftUpdate $False -AllowUseMeteredNetwork $False
```

Related topics

[Start-CMSoftwareUpdateDeployment](#)

Set-CMSoftwareUpdateDeploymentPackage

Set-CMSoftwareUpdateDeploymentPackage

Modifies a software update deployment package.

Syntax

Parameter Set: SetById

```
Set-CMSoftwareUpdateDeploymentPackage -Id <String[]> [-Description <String> ] [-NewName <String> ] [-Path <String> ] [-Priority {High | Low | Normal} ] [-SecuredScopeNames <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetByName

```
Set-CMSoftwareUpdateDeploymentPackage -Name <String> [-Description <String> ] [-NewName <String> ] [-Path <String> ] [-Priority {High | Low | Normal} ] [-SecuredScopeNames <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetValueMandatory

```
Set-CMSoftwareUpdateDeploymentPackage -InputObject <IResultObject> [-Description <String> ] [-NewName <String> ] [-Path <String> ] [-Priority {High | Low | Normal} ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetSecurityScopeById

```
Set-CMSoftwareUpdateDeploymentPackage -Id <String[]> -SecurityScopeAction {AddMembership | RemoveMembership} -SecurityScopeName <String> [-SecuredScopeNames <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetSecurityScopeByName

```
Set-CMSoftwareUpdateDeploymentPackage -Name <String> -SecurityScopeAction {AddMembership | RemoveMembership} -SecurityScopeName <String> [-SecuredScopeNames <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetSecurityScopeByValue

```
Set-CMSoftwareUpdateDeploymentPackage -InputObject <IResultObject> -SecurityScopeAction {AddMembership | RemoveMembership} -SecurityScopeName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Set-CMSoftwareUpdateDeploymentPackage** cmdlet modifies a software update deployment package. A software update deployment package contains one or more software updates for deployment to a collection of computers.

Parameters

-Description<String>

Specifies a description for the deployment package.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Id<String[]>

Specifies an array of identifiers for the deployment package.

Aliases	PackageId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a **CMSoftwareUpdateDeploymentPackage** object. To obtain an **CMSoftwareUpdateDeploymentPackage** object, use the **Get-
CMSoftwareUpdateDeploymentPackage** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies a name for the deployment package.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-NewName<String>

Specifies a new name for the deployment package.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Path<String>

Specifies a package source (URL) for the deployment package.

Aliases	PackageSourcePath
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Priority<Priorities>

Specifies a distribution priority for the deployment package.

The acceptable values for this parameter are:

High	
Low	
Normal	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SecuredScopeNames<String>

Specifies names of security scopes for the deployment package. The default value is Default.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SecurityScopeAction<SecurityScopeActionType>

Specifies an action for a security scope. Valid values are: AddMembership and RemoveMembership.

The acceptable values for this parameter are:

AddMembership	
RemoveMembership	

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SecurityScopeName<String>

Specifies the name of a security scope. A security scope name can be Default or the name of a custom security scope.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Update the name and description of a software update deployment package

This command sets a new name and description for the deployment package that has the ID ST10000C.

```
PS C:\> Set-CMSoftwareUpdateDeploymentPackage -Id "ST10000C" -Description "Deployment pack 107" -NewName "SDPTest"
```

Example 2: Add membership to a security scope of a software update deployment package

This command adds membership for the security scope named testScopeName.

```
PS C:\> Set-CMSoftwareUpdateDeploymentPackage -Name "DP107" -SecurityScopeAction AddMembership -SecurityScopeName "testScopeName"
```

Example 3: Remove membership from a security scope of a software update deployment package

This command removes membership for the security scope named testScopeName.

```
PS C:\> Set-CMSoftwareUpdateDeploymentPackage -Name "DP107" -SecurityScopeAction  
RemoveMembership -SecurityScopeName "testScopeName"
```

Related topics

[Get-CMSoftwareUpdateDeploymentPackage](#)

[Remove-CMSoftwareUpdateDeploymentPackage](#)

Set-CMSoftwareUpdateGroup

Set-CMSoftwareUpdateGroup

Changes configuration settings for software update groups in Configuration Manager.

Syntax

Parameter Set: SetById

```
Set-CMSoftwareUpdateGroup -Id <String[]> [-Description <String> ] [-NewName <String> ] [-SecuredScopeNames <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetByName

```
Set-CMSoftwareUpdateGroup -Name <String[]> [-Description <String> ] [-NewName <String> ] [-SecuredScopeNames <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetValueMandatory

```
Set-CMSoftwareUpdateGroup -InputObject <IResultObject> [-Description <String> ] [-NewName <String> ] [-SecuredScopeNames <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetSecurityScopeById

```
Set-CMSoftwareUpdateGroup -Id <String[]> -SecurityScopeAction {AddMembership | RemoveMembership} -SecurityScopeName <String> [-SecuredScopeNames <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetSecurityScopeByName

```
Set-CMSoftwareUpdateGroup -Name <String[]> -SecurityScopeAction {AddMembership | RemoveMembership} -SecurityScopeName <String> [-SecuredScopeNames <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetSecurityScopeByValue

```
Set-CMSoftwareUpdateGroup -InputObject <IResultObject> -SecurityScopeAction {AddMembership | RemoveMembership} -SecurityScopeName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Set-CMSoftwareUpdateGroup** cmdlet changes the name or description of one or more Microsoft System Center 2012 Configuration Manager software update groups, or it adds or removes software update groups for one or more security scopes.

A software update group is a collection of one or more software updates. You can add software updates to a software update group and then deploy the group to clients. After you deploy a software update group, you can add new software updates to the group and System Center 2012 Configuration Manager automatically deploys them.

Parameters

-Description<String>

Specifies a description for a software update group.

Aliases	LocalizedDescription
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Id<String[]>

Specifies an array of IDs of software update groups.

Aliases	CId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a software group object. To obtain a software group object, use the [Get-CMSoftwareUpdateGroup](#) cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-Name<String[]>

Specifies an array of names of software update groups.

Aliases	LocalizedDisplayName
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-NewName<String>

Specifies a name for a software update group. This name replaces the current name of the group.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SecuredScopeNames<String>

Specifies a security scope name.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-SecurityScopeAction<SecurityScopeActionType>

Specifies an action for a security scope. Valid values are: AddMembership and RemoveMembership.

The acceptable values for this parameter are:

AddMembership	
RemoveMembership	

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SecurityScopeName<String>

Specifies the name of a security scope. A security scope name can be Default or the name of a custom security scope.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Add a software update group to a security scope

This command adds a software update group named SUGroup01 as a member of the security scope named ScopeNameD02.

```
PS C:\> Set-CMSoftwareUpdateGroup -SecurityScopeAction AddMembership -SecurityScopeName "ScopeNameD02" -Name "SUGroup01"
```

Example 2: Remove a software update group from a security scope

This command removes the software update group named SUGroup01 from membership in the security scope named ScopeNameD17.

```
PS C:\> Set-CMSoftwareUpdateGroup -SecurityScopeAction RemoveMembership -SecurityScopeName "ScopeNameD17" -Name "SUGroup01"
```

Related topics

[Get-CMSoftwareUpdateGroup](#)

[New-CMSoftwareUpdateGroup](#)

[Remove-CMSoftwareUpdateGroup](#)

Set-CMSoftwareUpdatePoint

Set-CMSoftwareUpdatePoint

Changes settings for a Configuration Manager software update point.

Syntax

Parameter Set: SumPWithWsus

```
Set-CMSoftwareUpdatePoint -SiteCode <String> -SiteSystemServerName <String> [-AnonymousWsusAccess] [-ClientConnectionType {Internet | InternetAndIntranet | Intranet} ] [-UseProxy <Boolean> ] [-UseProxyForAutoDeploymentRule <Boolean> ] [-WsusAccessAccount <String> ] [-WsusiisPort <Int32> ] [-WsusiissslPort <Int32> ] [-WsusSsl <Boolean> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: NLBVIP

```
Set-CMSoftwareUpdatePoint -NlbVirtualIP <String> -SiteCode <String> -SiteSystemServerName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: NLBVIPByValue

```
Set-CMSoftwareUpdatePoint -InputObject <IResultObject> -NlbVirtualIP <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: PublicVIP

```
Set-CMSoftwareUpdatePoint -PublicVirtualIP <String> -SiteCode <String> -SiteSystemServerName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: PublicVIPByValue

```
Set-CMSoftwareUpdatePoint -InputObject <IResultObject> -PublicVirtualIP <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SumPByValueWithWsus

```
Set-CMSoftwareUpdatePoint -InputObject <IResultObject> [-AnonymousWsusAccess] [-ClientConnectionType {Internet | InternetAndIntranet | Intranet} ] [-UseProxy <Boolean> ] [-UseProxyForAutoDeploymentRule <Boolean> ] [-WsusAccessAccount <String> ] [-WsusiisPort <Int32> ] [-WsusiissslPort <Int32> ] [-WsusSsl <Boolean> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Set-CMSoftwareUpdatePoint** cmdlet changes settings for a software update point in Microsoft System Center 2012 Configuration Manager.

A software update point is a site server role that hosts software updates. System Center 2012 Configuration Manager clients connect to a software update point to get available updates. The

software update point interacts with Windows Server Update Services (WSUS) to configure update settings, request synchronization to the update source, and to synchronize software updates from the WSUS database.

You can use this cmdlet to configure the settings a software update point uses when connecting with clients and with a WSUS server. These settings include Network Load Balancing (NLB), a virtual IP address, Internet Information Services (IIS) port, and whether to use Secure Socket Locket Layer (SSL) to connect with WSUS.

Parameters

-AnonymousWsusAccess

Indicates that the software update point allows anonymous.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ClientConnectionType<ClientConnectionTypes>

Specifies a connection type. Clients can connect to the software update point in several ways. You can configure the software update point to handle different types of connections differently by specifying the connection type. Valid values are:

- Internet
- InternetAndIntranet
- Intranet

The acceptable values for this parameter are:

Internet	
InternetAndIntranet	
Intranet	

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a software update point object. To obtain a software update point object, use the **Get-
CMSoftwareUpdatePoint** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-NlbVirtualIP<String>

Specifies an IP address or host name. If this software update point uses load balancing, this is the NLB address.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PublicVirtualIP<String>

Specifies a public virtual IP address for a software update point that is connected to over the Internet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteCode<String>

Specifies a site code for a Configuration Manager site.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteSystemServerName<String>

Specifies the name of the computer that hosts the software update point site system role.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UseProxy<Boolean>

Specifies whether a software update point can use a proxy.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UseProxyForAutoDeploymentRule<Boolean>

Indicates whether an auto deployment rule can use a proxy.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-WsusAccessAccount<String>

Specifies an access account. Unless a software update point allows anonymous access, use this access account to connect to it.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-WsusiisPort<Int32>

Specifies a port to use for unsecured access to the WSUS server.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-WsusiSslPort<Int32>

Specifies a port to user for secured access to the WSUS server.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-WsusSsl<Boolean>

Specifies whether the software update point uses SSL to connect to the WSUS server.

Aliases	SslWsus
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Modify the server name

This command modifies the name for the site system server for the site code CM1.

```
PS C:\> Set-CMSoftwareUpdatePoint -SiteCode "CM1" -SiteSystemServerName  
"CMSoftwareUpdatePoint.Western.Contoso.com"
```

Related topics

[Add-CMSoftwareUpdatePoint](#)

[Get-CMSoftwareUpdatePoint](#)

[Remove-CMSoftwareUpdatePoint](#)

Set-CMSoftwareUpdatePointComponent

Set-CMSoftwareUpdatePointComponent

Modifies a software update point.

Syntax

Parameter Set: SearchBySiteCodeMandatory

```
Set-CMSoftwareUpdatePointComponent -SiteCode <String> [-AddLanguageSummaryDetails <String[]> ] [-AddLanguageUpdateFile <String[]> ] [-AddUpdateClassification <String[]> ] [-EnableSyncFailureAlert <Boolean> ] [-EnableSynchronization <Boolean> ] [-ImmediatelyExpireSupersedence <Boolean> ] [-RemoveLanguageSummaryDetails <String[]> ] [-RemoveLanguageUpdateFile <String[]> ] [-RemoveUpdateClassification <String[]> ] [-ReportingEvent {CreateAllWsusReportingEvents | CreateOnlyWsusStatusReportingEvents | DoNotCreateWsusReportingEvents} ] [-Schedule <IResultObject> ] [-SynchronizeAction {DoNotSynchronizeFromMicrosoftUpdateOrUpstreamDataSource | SynchronizeFromAnUpstreamDataSourceLocation | SynchronizeFromMicrosoftUpdate} ] [-UpstreamSourceLocation <String> ] [-WaitMonth <Int32> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByNameMandatory

```
Set-CMSoftwareUpdatePointComponent -Name <String> [-AddLanguageSummaryDetails <String[]> ] [-AddLanguageUpdateFile <String[]> ] [-AddUpdateClassification <String[]> ] [-EnableSyncFailureAlert <Boolean> ] [-EnableSynchronization <Boolean> ] [-ImmediatelyExpireSupersedence <Boolean> ] [-RemoveLanguageSummaryDetails <String[]> ] [-RemoveLanguageUpdateFile <String[]> ] [-RemoveUpdateClassification <String[]> ] [-ReportingEvent {CreateAllWsusReportingEvents | CreateOnlyWsusStatusReportingEvents | DoNotCreateWsusReportingEvents} ] [-Schedule <IResultObject> ] [-SynchronizeAction {DoNotSynchronizeFromMicrosoftUpdateOrUpstreamDataSource | SynchronizeFromAnUpstreamDataSourceLocation | SynchronizeFromMicrosoftUpdate} ] [-UpstreamSourceLocation <String> ] [-WaitMonth <Int32> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Set-CMSoftwareUpdatePointComponent -InputObject <IResultObject> [-AddLanguageSummaryDetails <String[]> ] [-AddLanguageUpdateFile <String[]> ] [-AddUpdateClassification <String[]> ] [-EnableSyncFailureAlert <Boolean> ] [-EnableSynchronization <Boolean> ] [-ImmediatelyExpireSupersedence <Boolean> ] [-RemoveLanguageSummaryDetails <String[]> ] [-RemoveLanguageUpdateFile <String[]> ] [-RemoveUpdateClassification <String[]> ] [-ReportingEvent {CreateAllWsusReportingEvents | CreateOnlyWsusStatusReportingEvents | DoNotCreateWsusReportingEvents} ] [-Schedule <IResultObject> ] [-SynchronizeAction {DoNotSynchronizeFromMicrosoftUpdateOrUpstreamDataSource | SynchronizeFromAnUpstreamDataSourceLocation | SynchronizeFromMicrosoftUpdate} ] [-UpstreamSourceLocation <String> ] [-WaitMonth <Int32> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Set-CMSoftwareUpdatePointComponent** cmdlet modifies a software update point. A software update point component interacts with a Windows Server Update Services (WSUS) server to configure update settings, request synchronization to the upstream update source, and synchronize updates from the WSUS database to the site server database on the central site.

You can specify a software update point to modify by name, by site code, or by using the **Get-CMSoftwareUpdatePointComponent** cmdlet.

Parameters

-AddLanguageSummaryDetails<String[]>

Specifies an array of languages, as strings. The cmdlet adds these languages to the languages supported for summary details at this site. Summary details information specifies the languages to download the metadata for the software updates during synchronization, such as name, description, products that the update supports, update classification, article ID, download URL, and applicability rules.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AddLanguageUpdateFile<String[]>

Specifies an array of languages, as strings. The cmdlet adds these languages to the languages supported for software updates at this site.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-AddUpdateClassification<String[]>

Specifies an array of software update classifications, as strings. This cmdlet adds these classifications to the classifications supported for software updates at this site.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnableSyncFailureAlert<Boolean>

Indicates whether Configuration Manager creates an alert when synchronization fails on a site.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnableSynchronization<Boolean>

Indicates whether this site automatically synchronizes updates according to a schedule. Specify a schedule by using the *Schedule* parameter.

Aliases	none
Required?	false
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ImmediatelyExpireSupersedence<Boolean>

Indicates whether a software update expires immediately after another update supersedes it or after a specified period of time. If you specify a value of `$False` for this parameter, specify the number of months to wait for expiration by using the *WaitMonth* parameter.

System Center 2012 Endpoint Protection definition updates and software updates that Service Packs supersede never expire.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a software update point component object. To obtain a software update point component object, use the **Get-CMSoftwareUpdatePointComponent** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies a name of a site system server in Configuration Manager.

Aliases	SiteName
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RemoveLanguageSummaryDetails<String[]>

Specifies an array of languages, as strings. The cmdlet removes these languages from the languages supported for summary details at this site. Summary details information specifies the languages to download the metadata for the software updates during synchronization, such as name, description, products that the update supports, update classification, article ID, download URL, and applicability rules.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RemoveLanguageUpdateFile<String[]>

Specifies an array of languages, as strings. The cmdlet removes these languages from the languages supported for software updates at this site.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RemoveUpdateClassification<String[]>

Specifies an array of software update classifications, as strings. This cmdlet removes these classifications from the classifications supported for software updates at this site.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ReportingEvent<ReportingEventType>

Specifies whether to create event messages for WSUS reporting for status reporting events or for all reporting events. Valid values are:

- CreateAllWsusReportingEvents
- CreateOnlyWsusStatusReportingEvents
- DoNotCreateWsusReportingEvents

The acceptable values for this parameter are:

CreateAllWsusReportingEvents	
CreateOnlyWsusStatusReportingEvents	
DoNotCreateWsusReportingEvents	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Schedule<IResultObject>

Specifies a **Schedule** object. Configuration Manager can synchronize updates according this schedule if you specify a value of \$True for the *EnableSynchronization* parameter. To obtain a **Schedule** object, use the **New-CMSchedule** cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteCode<String>

Specifies a site code in Configuration Manager.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SynchronizeAction<SynchronizeActionType>

Specifies a source for synchronization for this software update point. Valid values are:

- DoNotSynchronizeFromMicrosoftUpdateOrUpstreamDataSource
- SynchronizeFromAnUpstreamDataSourceLocation
- SynchronizeFromMicrosoftUpdate

If you select a value of *SynchronizeFromAnUpstreamDataSourceLocation*, specify the data source location by using the **UpstreamSourceLocation** parameter.

The acceptable values for this parameter are:

DoNotSynchronizeFromMicrosoftUpdateOrUpstreamDataSource	
---	--

SynchronizeFromAnUpstreamDataSourceLocation	
SynchronizeFromMicrosoftUpdate	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UpstreamSourceLocation<String>

Specifies an upstream data location as a URL. To use this location, specify a value of SynchronizeFromAnUpstreamDataSourceLocation for the *SynchronizeAction* parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-WaitMonth<Int32>

Specifies how long, in months, to wait before a software update expires after another update supersedes it. Specify a value of \$True for the *ImmediatelyExpireSupersedence* parameter for software updates to expire immediately.

Endpoint Protection definition updates and software updates that Service Packs supersede never expire.

Aliases	none
Required?	false
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Modify a software update point

The first command retrieves a software update point component object on the server named Contoso-SiteSysSrv.Western.Contoso.com, and stores the object in the **\$CIObj** variable.

The second command modifies the software update point component stored in **\$CIObj**.

```
PS C:\> $CIObj = Get-CMSoftwareUpdatePointComponent -SiteSystemServerName "Contoso-  
SiteSysSrv.Western.Contoso.com"
```

```
PS C:\> Set-CMSoftwareUpdatePointComponent -InputObject $CIObj
```

Related topics

[Get-CMSoftwareUpdatePointComponent](#)

[New-CMSchedule](#)

[Set-CMCollectionMembershipEvaluationComponent](#)

[Set-CMEmailNotificationComponent](#)

[Set-CMManagementPointComponent](#)

[Set-CMOutOfBandManagementComponent](#)

[Set-CMStatusReportingComponent](#)

[Set-CMSystemHealthValidatorPointComponent](#)

Set-CMSoftwareUpdateSummarizationSchedule

Set-CMSoftwareUpdateSummarizationSchedule

Sets how often Configuration Manager summarizes the status of updates.

Syntax

Parameter Set: Set

```
Set-CMSoftwareUpdateSummarizationSchedule -Interval <Int32> [-Unit {Days | Hours | Minutes}] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Set-CMSoftwareUpdateSummarizationSchedule** cmdlet sets how often Microsoft System Center 2012 Configuration Manager summarizes the status of software updates for all the System Center 2012 Configuration Manager sites. You can set the summary to run on an interval defined in days, hours, or minutes. You can use the **Invoke-CMSoftwareUpdateSummarization** cmdlet to run the summarization immediately.

Parameters

-Interval<Int32>

Specifies an amount of time, as an integer. This value works with the unit type you specify in the *Unit* parameter. Valid values for this parameter depend on the unit that you select:

- Minutes: 10 through 59.
- Hours: 1 through 23.
- Days: 1 through 31.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Unit<SummarizationScheduleUnit>

Specifies a unit to use to define an interval for the summarization schedule. Valid values are:

- Days
- Hours
- Minutes

The acceptable values for this parameter are:

Days	
Hours	
Minutes	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Schedule summarization interval and unit

This command sets the update summarization schedule to run every five days.

```
PS C:\> Set-CMSoftwareUpdateSummarizationSchedule -Interval 5 -Unit Days
```

Example 2: Change schedule interval

This command changes the interval for the update summarization schedule to seven. The command does not change the unit.

```
PS C:\> Set-CMSoftwareUpdateSummarizationSchedule -Interval 7
```

Related topics

[Get-CMSoftwareUpdateSummarizationSchedule](#)

[Invoke-CMSoftwareUpdateSummarization](#)

Set-CMStateMigrationPoint

Set-CMStateMigrationPoint

Modifies settings for a state migration point in Configuration Manager.

Syntax

Parameter Set: SetByName

```
Set-CMStateMigrationPoint -SiteCode <String> -SiteSystemServerName <String> [-  
AddBoundaryGroupName <String[]> ] [-AddStorageFolder <StorageDirectoryData[]> ] [-  
AllowFallbackSourceLocationForContent <Boolean> ] [-DeleteImmediately] [-  
EnableRestoreOnlyMode <Boolean> ] [-RemoveBoundaryGroupName <String[]> ] [-  
RemoveStorageFolder <StorageDirectoryData[]> ] [-TimeDeleteAfter <Int32> ] [-TimeUnit {Days  
| Hours} ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetByValue

```
Set-CMStateMigrationPoint -InputObject <IResultObject> [-AddBoundaryGroupName <String[]> ]  
[-AddStorageFolder <StorageDirectoryData[]> ] [-AllowFallbackSourceLocationForContent  
<Boolean> ] [-DeleteImmediately] [-EnableRestoreOnlyMode <Boolean> ] [-  
RemoveBoundaryGroupName <String[]> ] [-RemoveStorageFolder <StorageDirectoryData[]> ] [-  
TimeDeleteAfter <Int32> ] [-TimeUnit {Days | Hours} ] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Detailed Description

The **Set-CMStateMigrationPoint** cmdlet modifies settings for a state migration point in Microsoft System Center 2012 Configuration Manager. A state migration point is a site system role that manages data transfer from client computers during an operating system installation process. Use this cmdlet to modify the boundary groups and storage folders associated with the migration point, how long to wait before the migration point deletes client data, whether to allow a fallback source location for content, and whether to enable restore only mode.

You can specify which migration point to modify by using the site system server name and the site code, or use the **Get-CMStateMigrationPoint** cmdlet.

Parameters

-AddBoundaryGroupName<String[]>

Specifies an array of boundary group names. The cmdlet adds these boundary groups to the state migration point. During migration, clients in a boundary group use this site as a source location for content.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AddStorageFolder<StorageDirectoryData[]>

Specifies an array of storage folders, as storage directory data objects. The cmdlet adds these folders to the state migration point. To obtain a storage directory data object, use the **New-CMStorageFolder** cmdlet.

A state migration point stores user state data when it migrates a computer to a new operating system.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AllowFallbackSourceLocationForContent<Boolean>

Indicates whether a fallback source location is available.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeleteImmediately

Indicates that deletion of client data occurs immediately after the target computer downloads that data. If you select a value of `$False`, specify how long to wait by using the *TimeDeleteAfter* and *TimeUnit* parameters.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnableRestoreOnlyMode<Boolean>

Indicates whether to enable restore only mode. In restore only mode, Configuration Manager refuses new requests to store client data.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a state migration point object. To obtain a state migration point object, use the **Get-CMStateMigrationPoint** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-RemoveBoundaryGroupName<String[]>

Specifies an array of boundary group names. The cmdlet removes these boundary groups from the state migration point. During migration, clients in a boundary group use this site as a source location for content.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RemoveStorageFolder<StorageDirectoryData[]>

Specifies an array of storage folders, as storage directory data objects. The cmdlet removes these folders from the state migration point. A state migration point stores user state data when it migrates a computer to a new operating system.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteCode<String>

Specifies the site code for a Configuration Manager site.

Aliases	none
Required?	true

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteSystemServerName<String>

Specifies the host name for a state migration point.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-TimeDeleteAfter<Int32>

Specifies the amount of time to wait after the target computer downloads data to delete that data. Specify a time unit by using the *TimeUnit* parameter. To delete data immediately, specify a value of \$True for the *DeleteImmediately* parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-TimeUnit<IntervalType>

Specifies a time unit for the value specified in the *TimeDeleteAfter* parameter. Valid values are: Days and Hours.

The acceptable values for this parameter are:

Days	
Hours	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Modify a state migration point

This example modifies a migration point named MigrationServer.Western.Contoso.com for the site that has the code CM4. The example changes the migration point to allow a fallback source location for content, and modifies how long after data download to delete data.

The first command uses the **Get-CMStateMigrationPoint** cmdlet to obtain a migration point for the specified site code and server name, and stores it in the \$StateMigrationPoint variable.

The second command modifies the input object stored in the \$StateMigrationPoint variable. The command sets the *AllowFallbackSourceLocationForContent* parameter to \$True, and modifies the time to delete after to 12 hours.

```
PS C:\> $StateMigrationPoint = Get-CMStateMigrationPoint -SiteCode "CM4" -
SiteSystemServerName "MigrationServer.Western.Contoso.com"
PS C:\> Set-CMStateMigrationPoint -InputObject $StateMigrationPoint -
AllowFallbackSourceLocationForContent $True -TimeDeleteAfter 12 -TimeUnit Hours
```

Example 2: Modify storage folders and boundary groups for a state migration point

This example modifies settings for a state migration point named MigrationServer.Western.Contoso.com for the site that has the site code CM4. The example substitutes a different boundary group and different storage folder, and modifies other settings.

The first command uses the **New-CMStorageFolder** cmdlet to create a storage folder object, and stores it in the \$Storage01 variable. Consult documentation for that cmdlet for details.

The second command uses the **New-CMStorageFolder** cmdlet to create a storage folder object, and stores it in the \$Storage02 variable.

The third command removes the storage folder stored in the \$Storage01 variable from the migration point and, in the same command, adds the storage folder stored in the \$Storage02 variable to the migration point. Likewise, the command removes the boundary group named BG22 and adds the boundary group named BG07. The command also specifies a value of \$False for the *AllowFallbackSourceLocationForContent* parameter and a value of \$True for the *EnableRestoreOnlyMode* parameter. The command uses the *DeleteImmediately* parameter; therefore, the migration point deletes client information immediately after download.

```
PS C:\> $Storage01 = New-CMStorageFolder -MaximumClientNumber 100 -MinimumFreeSpace 100 -
SpaceUnit Megabyte -StorageFolderName "C:\"
PS C:\> $Storage02 = New-CMStorageFolder -MaximumClientNumber 100 -MinimumFreeSpace 10 -
SpaceUnit Gigabyte -StorageFolderName "D:\"
PS C:\> Set-CMStateMigrationPoint -SiteCode "CM4" -SiteSystemServerName
"MigrationServer.Western.Contoso.com" -AddBoundaryGroupName "BG07" -AddStorageFolder
```

```
$Storage02 -AllowFallbackSourceLocationForContent $False -DeleteImmediately -  
EnableRestoreOnlyMode $True -RemoveBoundaryGroupName "BG22" -RemoveStorageFolder $Storage01
```

Related topics

[Add-CMStateMigrationPoint](#)

[Get-CMStateMigrationPoint](#)

[Remove-CMStateMigrationPoint](#)

[New-CMStoragefolder](#)

Set-CMStatusFilterRule

Set-CMStatusFilterRule

Modifies settings for a Configuration Manager filter rule for status messages.

Syntax

Parameter Set: SetStatusFilterRulePriority

```
Set-CMStatusFilterRule -Name <String> -Priority {Decrease | Increase} -SiteCode <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetStatusFilterRuleProperty

```
Set-CMStatusFilterRule -Name <String> -SiteCode <String> [-AllowUserDeleteMessagesAfterThresholdDays <Int32> ] [-ComponentName <String> ] [-ForwardToStatusSummarizer <Boolean> ] [-MessageId <String> ] [-MessageType {Audit | Detail | Milestone | None} ] [-ProcessLowerPriorityRule <Boolean> ] [-ProgramPath <String> ] [-PropertyId <String> ] [-PropertyValue <String> ] [-ReplicateToParentSite <Boolean> ] [-ReplicationPriority {High | Low | Medium} ] [-ReportToEventLog <Boolean> ] [-RunProgram <Boolean> ] [-SeverityType {Error | Informational | None | Warning} ] [-SiteSystemServerName <String> ] [-Source <String> ] [-StatusFilterRuleSiteCode <String> ] [-WriteToDatabase <Boolean> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Set-CMStatusFilterRule** cmdlet modifies settings for a Microsoft System Center 2012 Configuration Manager filter rule for status messages. System Center 2012 Configuration Manager checks a status message against rules in order of priority. A rule can specify that rules with lower priority do not apply to a message after that rule applied.

Status filter rules specify how System Center 2012 Configuration Manager responds to status messages. Each filter rule contains criteria and actions for status messages. You configure status filter rules for each site, not across all sites.

To change the priority of a rule, use the rule name to specify the rule.

Parameters

-AllowUserDeleteMessagesAfterThresholdDays<Int32>

Specifies how long, in days, to retain messages. Specify a value of \$True for the *WriteToDatabase* parameter to specify a value for this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ComponentName<String>

Specifies the Configuration Manager component that corresponds to the status messages.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ForwardToStatusSummarizer<Boolean>

Indicates whether to forward to the status summarizer.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-MessageId<String>

Specifies a message ID in Configuration Manager.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-MessageType<MessageType>

Specifies a status message type in Configuration Manager. Valid values are: Audit, Detail, Milestone, and None.

The acceptable values for this parameter are:

Audit	
Detail	
Milestone	
None	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies an array of names for status filter rules.

Aliases	none
Required?	true
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Priority<PriorityChangeType>

Specifies a change in priority. Configuration Manager checks status messages against rules in order of rule priority. Valid values are: Decrease and Increase.

The acceptable values for this parameter are:

Decrease	
Increase	

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ProcessLowerPriorityRule<Boolean>

Indicates whether to process a lower priority rule, which prevents further rule processing.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ProgramPath<String>

Specifies a path to a program that runs when a status message matches the status filter rule.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PropertyId<String>

Specifies a property ID in Configuration Manager.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PropertyValue<String>

Specifies a value for the corresponding *PropertyId* parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ReplicateToParentSite<Boolean>

Indicates whether to pass a message to the parent site.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ReplicationPriority<ReplicationPriority>

Specifies a replication priority for sending status messages to the parent site. Valid values are: High, Low, and Medium.

The acceptable values for this parameter are:

High	
Low	
Medium	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ReportToEventLog<Boolean>

Indicates whether to report an event in the Windows event log.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RunProgram<Boolean>

Indicates whether to run a program when a status message matches a filter rule.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SeverityType<SeverityType>

Specifies the severity of a status message. Valid values are: Error, Informational, None, and Warning.

The acceptable values for this parameter are:

Error	
Informational	
None	
Warning	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-SiteCode<String>

Specifies the site code for a Configuration Manager site.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteSystemServerName<String>

Specifies the name of a site system server in Configuration Manager.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Source<String>

Specifies the status message source to match. The possible sources are the following:

- Client
- SMS Provider
- Site Server

Aliases	none
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-StatusFilterRuleSiteCode<String>

Specifies a site code for the site from which the status message originated.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-WriteToDatabase<Boolean>

Indicates whether to write a message to the database. Specify a value of \$True for this parameter to enable the *AllowUserDeleteMessagesAfterThresholdDays* parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Increase the priority of a rule

This command increases the priority of a filter rule that has the specified name in a site that has the site code CM1.

```
PS C:\> Set-CMStatusFilterRule -Name "Status change to critical" -SiteCode "CM1" -Priority Increase
```

Related topics

[Disable-CMStatusFilterRule](#)

[Enable-CMStatusFilterRule](#)

[Get-CMStatusFilterRule](#)

[New-CMStatusFilterRule](#)

[Remove-CMStatusFilterRule](#)

Set-CMStatusMessageQuery

Set-CMStatusMessageQuery

Changes settings or security scope or deletes messages for a Configuration Manager status message query.

Syntax

Parameter Set: DeleteMessageByIdMandatory

```
Set-CMStatusMessageQuery -DeleteMessage -Id <String> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: DeleteMessageByNameMandatory

```
Set-CMStatusMessageQuery -DeleteMessage -Name <String> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: DeleteMessageByObjectMandatory

```
Set-CMStatusMessageQuery -DeleteMessage -InputObject <IResultObject> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SetSecurityScopeById

```
Set-CMStatusMessageQuery -Id <String> -SecurityScopeAction {AddMembership |  
RemoveMembership} -SecurityScopeName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetSecurityScopeByName

```
Set-CMStatusMessageQuery -Name <String> -SecurityScopeAction {AddMembership |  
RemoveMembership} -SecurityScopeName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetSecurityScopeByValue

```
Set-CMStatusMessageQuery -InputObject <IResultObject> -SecurityScopeAction {AddMembership |  
RemoveMembership} -SecurityScopeName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetStatusMessageQueryByIdMandatory

```
Set-CMStatusMessageQuery -Id <String> [-Comments <String> ] [-Expression <String> ] [-  
NewName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetStatusMessageQueryByNameMandatory

```
Set-CMStatusMessageQuery -Name <String> [-Comments <String> ] [-Expression <String> ] [-  
NewName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetStatusMessageQueryByObjectMandatory

```
Set-CMStatusMessageQuery -InputObject <IResultObject> [-Comments <String> ] [-Expression  
<String> ] [-NewName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Set-CMStatusMessageQuery** cmdlet changes settings for a Microsoft System Center 2012 Configuration Manager status message query. Status message queries return status messages from a System Center 2012 Configuration Manager site database. You can modify a comment, a Windows Management Infrastructure (WMI) expression, or the name of a query.

You can use this cmdlet with the *DeleteMessage* parameter to delete messages that this query finds.

This cmdlet can also add or remove a security scope for a message query. Every status message query must belong to at least one security scope. For more information about security scopes, see [Planning for Security in Configuration Manager](http://go.microsoft.com/fwlink/?LinkId=268426) (<http://go.microsoft.com/fwlink/?LinkId=268426>) on TechNet.

You can specify a name or ID for a query or use the **Get-CMStatusMessageQuery** cmdlet to obtain a query.

Parameters

-Comments<String>

Specifies a comment for a query.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeleteMessage

Indicates that messages found by this query are deleted from the Configuration Manager database.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Expression<String>

Specifies an expression in WMI Query Language (WQL).

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Id<String>

Specifies an ID for a status message query.

Aliases	QueryId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a status message query object. To obtain a status message query object, use the **Get-
CMStatusMessageQuery** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies a name for a status message query.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-NewName<String>

Specifies a new name for a query.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SecurityScopeAction<SecurityScopeActionType>

Specifies an action for a security scope. Valid values are: AddMembership and RemoveMembership.

The acceptable values for this parameter are:

AddMembership	
RemoveMembership	

Aliases	none
Required?	true
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SecurityScopeName<String>

Specifies the name of a security scope. A security scope name can be Default or the name of a custom security scope.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Add a security scope

This command adds the security scope named Scope22 to the query named All Status Messages.

```
PS C:\> Set-CMStatusMessageQuery -Name "All Status Messages" -SecurityScopeAction  
AddMembership -SecurityScopeName "Scope22"
```

Example 2: Delete messages

This command removes messages found by the query named All Active Directory Security Groups from the System Center 2012 Configuration Manager database.

```
PS C:\> Set-CMStatusMessageQuery -DeleteMessage -Name "All Active Directory Security Groups"
```

Example 3: Rename a query

This command renames the query All Active Directory Security Groups. The new name of the query is Western Security Groups.

```
PS C:\> Set-CMStatusMessageQuery -Name "All Active Directory Security Groups" -NewName  
"Western Security Groups"
```

Related topics

[Get-CMStatusMessageQuery](#)

[New-CMStatusMessageQuery](#)

[Remove-CMStatusMessageQuery](#)

Set-CMStatusReportingComponent

Set-CMStatusReportingComponent

Sets an object representing a status reporting component.

Syntax

Parameter Set: SearchBySiteCodeMandatory

```
Set-CMStatusReportingComponent -SiteCode <String> [-ClientLogChecked <Boolean> ] [-ClientLogFailureChecked <Boolean> ] [-ClientLogType {AllMilestones | AllMilestonesAndAllDetails | ErrorAndWarningMilestones | ErrorMilestones} ] [-ClientReportChecked <Boolean> ] [-ClientReportFailureChecked <Boolean> ] [-ClientReportType {AllMilestones | AllMilestonesAndAllDetails | ErrorAndWarningMilestones | ErrorMilestones} ] [-ServerLogChecked <Boolean> ] [-ServerLogFailureChecked <Boolean> ] [-ServerLogType {AllMilestones | AllMilestonesAndAllDetails | ErrorAndWarningMilestones | ErrorMilestones} ] [-ServerReportChecked <Boolean> ] [-ServerReportFailureChecked <Boolean> ] [-ServerReportType {AllMilestones | AllMilestonesAndAllDetails | ErrorAndWarningMilestones | ErrorMilestones} ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByNameMandatory

```
Set-CMStatusReportingComponent -Name <String> [-ClientLogChecked <Boolean> ] [-ClientLogFailureChecked <Boolean> ] [-ClientLogType {AllMilestones | AllMilestonesAndAllDetails | ErrorAndWarningMilestones | ErrorMilestones} ] [-ClientReportChecked <Boolean> ] [-ClientReportFailureChecked <Boolean> ] [-ClientReportType {AllMilestones | AllMilestonesAndAllDetails | ErrorAndWarningMilestones | ErrorMilestones} ] [-ServerLogChecked <Boolean> ] [-ServerLogFailureChecked <Boolean> ] [-ServerLogType {AllMilestones | AllMilestonesAndAllDetails | ErrorAndWarningMilestones | ErrorMilestones} ] [-ServerReportChecked <Boolean> ] [-ServerReportFailureChecked <Boolean> ] [-ServerReportType {AllMilestones | AllMilestonesAndAllDetails | ErrorAndWarningMilestones | ErrorMilestones} ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Set-CMStatusReportingComponent -InputObject <IResultObject> [-ClientLogChecked <Boolean> ] [-ClientLogFailureChecked <Boolean> ] [-ClientLogType {AllMilestones | AllMilestonesAndAllDetails | ErrorAndWarningMilestones | ErrorMilestones} ] [-ClientReportChecked <Boolean> ] [-ClientReportFailureChecked <Boolean> ] [-ClientReportType {AllMilestones | AllMilestonesAndAllDetails | ErrorAndWarningMilestones | ErrorMilestones} ] [-ServerLogChecked <Boolean> ] [-ServerLogFailureChecked <Boolean> ] [-ServerLogType {AllMilestones | AllMilestonesAndAllDetails | ErrorAndWarningMilestones | ErrorMilestones} ] [-ServerReportChecked <Boolean> ] [-ServerReportFailureChecked <Boolean> ] [-ServerReportType {AllMilestones | AllMilestonesAndAllDetails | ErrorAndWarningMilestones | ErrorMilestones} ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Set-CMStatusReportingComponent** cmdlet sets an object that represents a status reporting component. A status reporting component object specifies information about the client configuration and server configuration components.

You can configure the reporting component to check log files and monitor the severity of entries in the log files.

Parameters

-ClientLogChecked<Boolean>

Indicates whether a client log is checked.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ClientLogFailureChecked<Boolean>

Indicates whether a client log failure is checked.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ClientLogType<StatusReportOrLogType>

Specifies a type of client log. Valid values are:

- AllMilestones
- AllMilestonesAndAllDetails

-- AllMilestonesAndAllDetails

-- ErrorMilestones

The acceptable values for this parameter are:

AllMilestones	
AllMilestonesAndAllDetails	
ErrorAndWarningMilestones	
ErrorMilestones	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ClientReportChecked<Boolean>

Indicates whether a client report is checked.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ClientReportFailureChecked<Boolean>

Indicates whether a client failure is checked.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ClientReportType<StatusReportOrLogType>

Specifies a type of report. Valid values are:

- AllMilestones
- AllMilestonesAndAllDetails
- ErrorAndWarningMilestones
- ErrorMilestones

The acceptable values for this parameter are:

AllMilestones	
AllMilestonesAndAllDetails	
ErrorAndWarningMilestones	
ErrorMilestones	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies an object output from another cmdlet.

Aliases	none
---------	------

Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies a name for a status reporting component in Configuration Manager.

Aliases	SiteName
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ServerLogChecked<Boolean>

Indicates whether a server log is checked.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ServerLogFailureChecked<Boolean>

Indicates whether a server log failure is checked.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ServerLogType<StatusReportOrLogType>

Specifies a server log type. Valid values are:

- AllMilestones
- AllMilestonesAndAllDetails
- ErrorAndWarningMilestones
- ErrorMilestones

The acceptable values for this parameter are:

AllMilestones	
AllMilestonesAndAllDetails	
ErrorAndWarningMilestones	
ErrorMilestones	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ServerReportChecked<Boolean>

Indicates whether a server report is checked.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ServerReportFailureChecked<Boolean>

Indicates whether a server report failure is checked.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ServerReportType<StatusReportOrLogType>

Specifies a report type. Valid values are:

- AllMilestones
- AllMilestonesAndAllDetails
- ErrorAndWarningMilestones
- ErrorMilestones

The acceptable values for this parameter are:

AllMilestones	
AllMilestonesAndAllDetails	
ErrorAndWarningMilestones	
ErrorMilestones	

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteCode<String>

Specifies a site code for a Microsoft System Center 2012 Configuration Manager site.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Set status reporting component

This command sets a client report type and a server report type.

```
PS C:\> Set-CMStatusReportingComponent -SiteCode "CM1" -ClientReportType AllMilestones -  
ServerReportType AllMilestones
```

Related topics

[Get-CMStatusReportingComponent](#)

Set-CMStatusSummarizer

Set-CMStatusSummarizer

Modifies settings of a Configuration Manager status summarizer.

Syntax

Parameter Set: SetAppDeploymentSummarizer

```
Set-CMStatusSummarizer -ApplicationDeploymentSummarizer -SiteCode <String> [-DayInterval <Int32> ] [-HourInterval <Int32> ] [-MinuteInterval <Int32> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetAppStatisticsSummarizer

```
Set-CMStatusSummarizer -ApplicationStatisticsSummarizer -SiteCode <String> [-DayInterval <Int32> ] [-HourInterval <Int32> ] [-MinuteInterval <Int32> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetComponentStatusSummarizer

```
Set-CMStatusSummarizer -ComponentStatusSummarizer -SiteCode <String> [-EnableStatusSummarizer <Boolean> ] [-ReplicateToParentSite <Boolean> ] [-ReplicationPriority {High | Low | Normal} ] [-ThresholdPeriod <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetSiteSystemStatusSummarizer

```
Set-CMStatusSummarizer -SiteCode <String> -SiteSystemStatusSummarizer [-CriticalSizeKB <Int32> ] [-EnableStatusSummarizer <Boolean> ] [-ReplicateToParentSite <Boolean> ] [-ReplicationPriority {High | Low | Normal} ] [-Schedule <IResultObject> ] [-WarningSizeKB <Int32> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Set-CMStatusSummarizer** cmdlet modifies settings of a status summarizer. The Microsoft System Center 2012 Configuration Manager status summarizers apply to the areas of application deployment, application statistics, component status, and site system status.

Parameters

-ApplicationDeploymentSummarizer

Indicates that the summarizer is an application deployment status summarizer.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ApplicationStatisticsSummarizer

Indicates that the summarizer is an application statistics status summarizer.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ComponentStatusSummarizer

Indicates that the summarizer is a component status summarizer.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CriticalSizeKB<Int32>

Specifies the threshold, in kilobytes, of free space for a Critical message. This parameter applies to site system status summarizers.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DayInterval<Int32>

Specifies the frequency of status updates, in days, for a deployment last modified more than 90 days previously. This parameter is valid for application deployment status summarizers and application statistics summarizers.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnableStatusSummarizer<Boolean>

Indicates whether to enable Configuration Manager to summarize status messages for the site. This parameter applies to component status summarizers and site system status summarizers.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-HourInterval<Int32>

Specifies the frequency of status updates, in hours, for a deployment last modified in the last 31 to 90 days. This parameter is valid for application deployment status summarizers and application statistics summarizers.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-MinuteInterval<Int32>

Specifies the frequency of status updates, in minutes, for a deployment last modified in the last 30 days. This parameter is valid for application deployment status summarizers and application statistics summarizers.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ReplicateToParentSite<Boolean>

Indicates whether to pass summarization data from this site to its parent site. This parameter applies to component status summarizers and site system status summarizers. If you specify a value of \$True for this parameter, specify a priority by using the *ReplicationPriority* parameter and a threshold period by using the *ThresholdPeriod* parameter.

Aliases	none
Required?	false
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ReplicationPriority<ReplicationPriorityType>

Specifies the priority of replication to the parent site. Valid values are: Low, Normal, and High. Specify this value if you specified \$True for the *ReplicateToParentSite* parameter.

The acceptable values for this parameter are:

High	
Low	
Normal	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Schedule<IResultObject>

Specifies a schedule object that determines how often to summarize site system status. To obtain a schedule object, use the **New-CMSchedule** cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteCode<String>

Specifies the site code for a Configuration Manager site.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteSystemStatusSummarizer

Indicates that the summarizer is a site system status summarizer.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ThresholdPeriod<String>

Specifies a threshold period for replicating summarization information to a parent site. Specify a value if you specified a value of \$True for the *ReplicateToParentSite* parameter. This parameter applies to component status summarizers.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-WarningSizeKB<Int32>

Specifies the threshold, in kilobytes, of free space for a Warning message. This parameter applies to site system status summarizers.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

[New-CMSchedule](#)

[Get-CMStatusSummarizer](#)

Set-CMSystemHealthValidatorPointComponent

Set-CMSystemHealthValidatorPointComponent

Modifies settings of a Configuration Manager system health validator point.

Syntax

Parameter Set: SearchBySiteCodeMandatory

```
Set-CMSystemHealthValidatorPointComponent -SiteCode <String> [-Date <DateTime> ] [-DesignateActiveDirectoryForest <Boolean> ] [-DomainSuffix <String> ] [-PublishAccount <String> ] [-QueryAccount <String> ] [-QueryInterval <Int32> ] [-Time <DateTime> ] [-UseDateTime <Boolean> ] [-ValidityPeriod <Int32> ] [-Confirm] [-WhatIf] [<CommonParameters>]
```

Parameter Set: SearchByNameMandatory

```
Set-CMSystemHealthValidatorPointComponent -Name <String> [-Date <DateTime> ] [-DesignateActiveDirectoryForest <Boolean> ] [-DomainSuffix <String> ] [-PublishAccount <String> ] [-QueryAccount <String> ] [-QueryInterval <Int32> ] [-Time <DateTime> ] [-UseDateTime <Boolean> ] [-ValidityPeriod <Int32> ] [-Confirm] [-WhatIf] [<CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Set-CMSystemHealthValidatorPointComponent -InputObject <IResultObject> [-Date <DateTime> ] [-DesignateActiveDirectoryForest <Boolean> ] [-DomainSuffix <String> ] [-PublishAccount <String> ] [-QueryAccount <String> ] [-QueryInterval <Int32> ] [-Time <DateTime> ] [-UseDateTime <Boolean> ] [-ValidityPeriod <Int32> ] [-Confirm] [-WhatIf] [<CommonParameters>]
```

Detailed Description

The **Set-CMSystemHealthValidatorPointComponent** cmdlet modifies settings of a system health validator point. A system health validator point is a Microsoft System Center 2012 Configuration Manager site system role that evaluates system health information reported by Windows clients for security related compliance.

You can modify whether the system health validator point uses the current Active Directory forest or a designated forest. You can specify the accounts that the component uses to publish and query Active Directory Domain Services (AD DS). You can set the validity period for cached statements of health and whether to accept statements of health only after a specific time. Any changes you make apply to all system health validator points in the System Center 2012 Configuration Manager site.

To specify a system health validator point to modify, specify a site code or name, or you can use the **Get-CMSystemHealthValidatorPointComponent** cmdlet to get a system health validator point to modify.

Parameters

-Date<DateTime>

Specifies a date, as a **DateTime** object. To obtain a **DateTime** object, use the **Get-Date** cmdlet. For more information, type `Get-Help Get-Date`.

If you specify a value of `$True` for the *UseDateTime* parameter, a client must create a statement of health after the date and time specified by using this parameter and the *Time* parameter. The date and time must be in the past.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DesignateActiveDirectoryForest<Boolean>

Indicates whether the site system server and the system health validator points are in different Active Directory forests. If the value is `$True`, specify an Active Directory forest by using the *DomainSuffix* parameter. If no trust relationship exists between the forests, you may need to specify accounts by using the *PublishAccount* and *QueryAccount* parameters.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DomainSuffix<String>

Specifies a domain suffix for a designated Active Directory forest. If no trust relationship exists between this forest and the site system server forest, you may need to specify accounts by using the *PublishAccount* and *QueryAccount* parameters.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a system health validator point object. To obtain a system health validator point, use the **Get-CMSystemHealthValidatorPointComponent** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies the name of a system health validator point.

Aliases	SiteName
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PublishAccount<String>

Specifies a health state reference publishing account, in the format *Domain\User*. If you do not specify an account, the component uses the site system server account.

You must specify an account if no trust relationship exists between the site server domain and the domain suffix specified in the *DomainSuffix* parameter or if there is a trust relationship, but the site system server account lacks Full Control permission for the System Management Active Directory container.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-QueryAccount<String>

Specifies an account, in the format *Domain\User*, that the system health validator point uses to query AD DS for state references. If you do not specify an account, the component uses the site system server account.

You must specify an account if no trust relationship exists between the site server domain and the domain suffix specified in the *DomainSuffix* parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-QueryInterval<Int32>

Specifies how often, in minutes, system health validator points get health state references. Valid values are: 1 to 10080 minutes (7 days). The default value is 120 minutes.

Aliases	none
Required?	false
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteCode<String>

Specifies a site code in Configuration Manager.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Time<DateTime>

Specifies a time, as a **DateTime** object. To obtain a **DateTime** object, use the **Get-Date** cmdlet.

If you specify a value of \$True for the *UseDateTime* parameter, a client must create a statement of health after the date and time specified by using this parameter and the *Date* parameter. The date and time must be in the past.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UseDateTime<Boolean>

Indicates whether a client must create a statement of health after a specific date and time. If you select a value of \$True, specify the date and time by using the *Date* and *Time* parameters. The date and time must be in the past. The default value for this parameter is \$False.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ValidityPeriod<Int32>

Specifies the length of time, in hours, for which system health validator points accept a cached client statement of health. Valid values are: 1 to 168 hours (7 days). The default value is 26 hours.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Modify settings of a system health validator point by using a name

This command modifies settings of a system health validator point named SHVPC02.Western.Contoso.com. The command changes the query interval to 60 minutes and the validity period to 24 hours.

```
PS C:\> Set-CMSystemHealthValidatorPointComponent -Name "SHVPC02.Western.Contoso.com" -QueryInterval 60 -ValidityPeriod 24
```

Related topics

[Get-CMSystemHealthValidatorPointComponent](#)

[Set-CMCollectionMembershipEvaluationComponent](#)

[Set-CMEmailNotificationComponent](#)

[Set-CMManagementPointComponent](#)

[Set-CMOutOfBandManagementComponent](#)

[Set-CMSoftwareUpdatePointComponent](#)

[Set-CMStatusReportingComponent](#)

Set-CMTaskSequence

Set-CMTaskSequence

Modifies a Configuration Manager task sequence.

Syntax

Parameter Set: SetById

```
Set-CMTaskSequence -TaskSequenceId <String> [-BootImageId <String> ] [-Category <String> ] [-CustomText <String> ] [-DeploymentPackageId <String> ] [-Description <String> ] [-DisableTaskSequence <Boolean> ] [-Duration <Int64> ] [-NewName <String> ] [-ProgramName <String> ] [-RunAnotherProgram <Boolean> ] [-RunEveryTime <Boolean> ] [-SuppressNotification <Boolean> ] [-UseBootImage <Boolean> ] [-UseDefaultText <Boolean> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetByValue

```
Set-CMTaskSequence -TaskSequence <IResultObject> [-BootImageId <String> ] [-Category <String> ] [-CustomText <String> ] [-DeploymentPackageId <String> ] [-Description <String> ] [-DisableTaskSequence <Boolean> ] [-Duration <Int64> ] [-NewName <String> ] [-ProgramName <String> ] [-RunAnotherProgram <Boolean> ] [-RunEveryTime <Boolean> ] [-SuppressNotification <Boolean> ] [-UseBootImage <Boolean> ] [-UseDefaultText <Boolean> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetSecurityScopeById

```
Set-CMTaskSequence -SecurityScopeAction {AddMembership | RemoveMembership} - SecurityScopeName <String> -TaskSequencePackageId <String[]> [-SecuredScopeNames <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetSecurityScopeByName

```
Set-CMTaskSequence -Name <String> -SecurityScopeAction {AddMembership | RemoveMembership} - SecurityScopeName <String> [-SecuredScopeNames <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetSecurityScopeByValue

```
Set-CMTaskSequence -InputObject <IResultObject> -SecurityScopeAction {AddMembership | RemoveMembership} -SecurityScopeName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Set-CMTaskSequence** cmdlet modifies a Microsoft System Center 2012 Configuration Manager task sequence. A task sequence includes configuration and operating system deployment settings for a System Center 2012 Configuration Manager client computer.

To modify security scopes, specify a name or ID for a sequence or use the **Get-CMTaskSequence** cmdlet to obtain a sequence. To change a sequence that has a particular security scope, you can specify a security scope along with a name or ID.

To modify other settings, specify an ID or use the **Get-CMTaskSequence** cmdlet to obtain a sequence.

Parameters

-BootImageId<String>

Specifies the ID of a boot image that the task sequence references.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Category<String>

Specifies a category, as a string, for the task sequence. You can use categories to group task sequences.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CustomText<String>

Specifies custom text for the task sequence. Custom text appears in the progress notification dialog box while the task sequence runs. The text cannot exceed 50 characters.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeploymentPackageId<String>

Specifies the ID of a package. If you specify a value of \$True for the *RunAnotherProgram* parameter, the specified package runs before the task sequence runs.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Description<String>

Specifies a description for the task sequence.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DisableTaskSequence<Boolean>

Indicates whether to disable this task sequence.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Duration<Int64>

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a task sequence object. To obtain a task sequence object, use the **Get-CMTaskSequence** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies a name of a task sequence.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-NewName<String>

Specifies a new name for the task sequence.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ProgramName<String>

Specifies the name of a program to run from a Configuration Manager software package specified by the *DeploymentPackageId* parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RunAnotherProgram<Boolean>

Indicates whether to run another program before running the task sequence. Specify the program by using the *DeploymentPackageId* parameter and the *ProgramName* parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RunEveryTime<Boolean>

Indicates whether the program specified in the *ProgramName* parameter runs every time that the task sequence runs. If you specify a value of `$False`, the program does not run if it has run successfully in the past.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SecuredScopeNames<String>

Specifies a name of a security scope. This parameter refers to the security scopes that a task sequence already belongs to, not a new scope to add.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SecurityScopeAction<SecurityScopeActionType>

Specifies an action for a security scope. Valid values are: AddMembership and RemoveMembership.
The acceptable values for this parameter are:

AddMembership	
RemoveMembership	

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SecurityScopeName<String>

Specifies the name of a security scope. A security scope name can be Default or the name of a custom security scope.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SuppressNotification<Boolean>

Indicates whether to suppress notifications for this task sequence.

Aliases	none
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-TaskSequence<IResultObject>

Specifies a task sequence object. To obtain a task sequence object, use the **Get-CMTaskSequence** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-TaskSequenceId<String>

Specifies the ID of a task sequence.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-TaskSequencePackageId<String[]>

Specifies an array of IDs of task sequences.

Aliases	PackageId
---------	-----------

Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UseBootImage<Boolean>

Indicates whether the task sequence uses the boot image specified by using the *BootImageID* parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UseDefaultText<Boolean>

Indicates whether to use default text in the progress notification dialog box while the task sequence runs. If you select a value of *\$False* for this parameter, be sure to specify custom text by using the *CustomText* parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Add a security scope to a task sequence

This command adds the specified security scope to the task sequence named General Sequence 22.

```
PS C:\> Set-CMTaskSequence -Name "General Sequence 22" -SecurityScopeAction AddMembership -SecurityScopeName "Scope05"
```

Related topics

[Disable-CMTaskSequence](#)

[Enable-CMTaskSequence](#)

[Export-CMTaskSequence](#)

[Get-CMTaskSequence](#)

[Import-CMTaskSequence](#)

[New-CMTaskSequence](#)

[Remove-CMTaskSequence](#)

Set-CMTaskSequenceDeployment

Set-CMTaskSequenceDeployment

Creates a task sequence deployment in Configuration Manager.

Syntax

Parameter Set: SetTaskSequenceDeploymentByDeploymentIdMandatory

```
Set-CMTaskSequenceDeployment -AllowUseRemoteDistributionPoint <Boolean> -DeploymentOption {DownloadAllContentLocallyBeforeStartingTaskSequence | DownloadContentLocallyWhenNeededByRunningTaskSequence} -TaskSequenceDeploymentId <String> [-AlertDay <DateTime> ] [-AlertTime <DateTime> ] [-AllowFallback <Boolean> ] [-AllowUsersRunIndependently <Boolean> ] [-Comment <String> ] [-CreateAlertBaseOnPercentFailure <Boolean> ] [-CreateAlertBaseOnPercentSuccess <Boolean> ] [-DeploymentAvailableDay <DateTime> ] [-DeploymentAvailableTime <DateTime> ] [-DeploymentExpireDay <DateTime> ] [-DeploymentExpireTime <DateTime> ] [-InternetOption <Boolean> ] [-MakeAvailableTo <MakeAvailableToType> ] [-PercentFailure <Int32> ] [-PercentSuccess <Int32> ] [-PersistOnWriteFilterDevice <Boolean> ] [-RerunBehavior {AlwaysRerunProgram | NeverRerunDeployedProgram | RerunIfFailedPreviousAttempt | RerunIfSucceededOnPreviousAttempt} ] [-Schedule <IResultObject[]> ] [-ScheduleEvent {AsSoonAsPossible | LogOff | LogOn} ] [-SendWakeUpPacket <Boolean> ] [-ShowTaskSequenceProgress <Boolean> ] [-SoftwareInstallation <Boolean> ] [-SystemRestart <Boolean> ] [-UseMeteredNetwork <Boolean> ] [-UseUtcForAvailableSchedule <Boolean> ] [-UseUtcForExpireSchedule <Boolean> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetTaskSequenceDeploymentByIdMandatory

```
Set-CMTaskSequenceDeployment -AllowUseRemoteDistributionPoint <Boolean> -CollectionName <String> -DeploymentOption {DownloadAllContentLocallyBeforeStartingTaskSequence | DownloadContentLocallyWhenNeededByRunningTaskSequence} -TaskSequencePackageId <String> [-AlertDay <DateTime> ] [-AlertTime <DateTime> ] [-AllowFallback <Boolean> ] [-AllowUsersRunIndependently <Boolean> ] [-Comment <String> ] [-CreateAlertBaseOnPercentFailure <Boolean> ] [-CreateAlertBaseOnPercentSuccess <Boolean> ] [-DeploymentAvailableDay <DateTime> ] [-DeploymentAvailableTime <DateTime> ] [-DeploymentExpireDay <DateTime> ] [-DeploymentExpireTime <DateTime> ] [-InternetOption <Boolean> ] [-MakeAvailableTo <MakeAvailableToType> ] [-PercentFailure <Int32> ] [-PercentSuccess <Int32> ] [-PersistOnWriteFilterDevice <Boolean> ] [-RerunBehavior {AlwaysRerunProgram | NeverRerunDeployedProgram | RerunIfFailedPreviousAttempt | RerunIfSucceededOnPreviousAttempt} ] [-Schedule <IResultObject[]> ] [-ScheduleEvent {AsSoonAsPossible | LogOff | LogOn} ] [-SendWakeUpPacket <Boolean> ] [-ShowTaskSequenceProgress <Boolean> ] [-SoftwareInstallation <Boolean> ] [-SystemRestart <Boolean> ] [-UseMeteredNetwork <Boolean> ] [-UseUtcForAvailableSchedule <Boolean> ] [-UseUtcForExpireSchedule <Boolean> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetTaskSequenceDeploymentByNameMandatory

```
Set-CMTaskSequenceDeployment -AllowUseRemoteDistributionPoint <Boolean> -CollectionName <String> -DeploymentOption {DownloadAllContentLocallyBeforeStartingTaskSequence | DownloadContentLocallyWhenNeededByRunningTaskSequence} -TaskSequenceName <String> [-AlertDay
```

```

<DateTime> ] [-AlertTime <DateTime> ] [-AllowFallback <Boolean> ] [-
AllowUsersRunIndependently <Boolean> ] [-Comment <String> ] [-
CreateAlertBaseOnPercentFailure <Boolean> ] [-CreateAlertBaseOnPercentSuccess <Boolean> ] [-
DeploymentAvailableDay <DateTime> ] [-DeploymentAvailableTime <DateTime> ] [-
DeploymentExpireDay <DateTime> ] [-DeploymentExpireTime <DateTime> ] [-InternetOption
<Boolean> ] [-MakeAvailableTo <MakeAvailableToType> ] [-PercentFailure <Int32> ] [-
PercentSuccess <Int32> ] [-PersistOnWriteFilterDevice <Boolean> ] [-RerunBehavior
{AlwaysRerunProgram | NeverRerunDeployedProgram | RerunIfFailedPreviousAttempt |
RerunIfSucceededOnPreviousAttempt} ] [-Schedule <IResultObject[]> ] [-ScheduleEvent
{AsSoonAsPossible | LogOff | LogOn} ] [-SendWakeUpPacket <Boolean> ] [-
ShowTaskSequenceProgress <Boolean> ] [-SoftwareInstallation <Boolean> ] [-SystemRestart
<Boolean> ] [-UseMeteredNetwork <Boolean> ] [-UseUtcForAvailableSchedule <Boolean> ] [-
UseUtcForExpireSchedule <Boolean> ] [-Confirm] [-WhatIf] [ <CommonParameters>]

```

```

Parameter Set: SetTaskSequenceDeploymentByValueMandatory
Set-CMTaskSequenceDeployment -AllowUseRemoteDistributionPoint <Boolean> -CollectionName
<String> -DeploymentOption {DownloadAllContentLocallyBeforeStartingTaskSequence |
DownloadContentLocallyWhenNeededByRunningTaskSequence} -TaskSequence <IResultObject> [-
AlertDay <DateTime> ] [-AlertTime <DateTime> ] [-AllowFallback <Boolean> ] [-
AllowUsersRunIndependently <Boolean> ] [-Comment <String> ] [-
CreateAlertBaseOnPercentFailure <Boolean> ] [-CreateAlertBaseOnPercentSuccess <Boolean> ] [-
DeploymentAvailableDay <DateTime> ] [-DeploymentAvailableTime <DateTime> ] [-
DeploymentExpireDay <DateTime> ] [-DeploymentExpireTime <DateTime> ] [-InternetOption
<Boolean> ] [-MakeAvailableTo <MakeAvailableToType> ] [-PercentFailure <Int32> ] [-
PercentSuccess <Int32> ] [-PersistOnWriteFilterDevice <Boolean> ] [-RerunBehavior
{AlwaysRerunProgram | NeverRerunDeployedProgram | RerunIfFailedPreviousAttempt |
RerunIfSucceededOnPreviousAttempt} ] [-Schedule <IResultObject[]> ] [-ScheduleEvent
{AsSoonAsPossible | LogOff | LogOn} ] [-SendWakeUpPacket <Boolean> ] [-
ShowTaskSequenceProgress <Boolean> ] [-SoftwareInstallation <Boolean> ] [-SystemRestart
<Boolean> ] [-UseMeteredNetwork <Boolean> ] [-UseUtcForAvailableSchedule <Boolean> ] [-
UseUtcForExpireSchedule <Boolean> ] [-Confirm] [-WhatIf] [ <CommonParameters>]

```

Detailed Description

The **Set-CMTaskSequenceDeployment** cmdlet creates a task sequence deployment. A task sequence deployment assigns a task sequence to a collection of computers.

Parameters

-AlertDay<DateTime>

Specifies a day, in MM/DD/YYYY format, to trigger alerts. If you configure a percent success or failure rate for a deployment, alerts appear after this date.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AlertTime<DateTime>

Specifies a time, in HH:MM format, to trigger alerts. If you configure a percent success or failure rate for a deployment, alerts appear after this time.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AllowFallback<Boolean>

Indicates whether to allow clients to use a fallback source location for content.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AllowUseRemoteDistributionPoint<Boolean>

Aliases	none
---------	------

Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AllowUsersRunIndependently<Boolean>

Indicates whether to allow users to independently run the program, regardless of its assignment status.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CollectionName<String>

Specifies a name of a collection designated to receive a task sequence deployment. A collection is a group of client computers.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Comment<String>

Specifies a comment for the task sequence deployment.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CreateAlertBaseOnPercentFailure<Boolean>

Indicates whether to create an alert, depending on the percentage of an installation that fails. If you specify a percent failure for installations, Configuration Manager raises an alert in the console when deployment fails on that percentage of installations. Use this parameter in conjunction with the *PercentFailure* parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CreateAlertBaseOnPercentSuccess<Boolean>

Indicates whether to create an alert, depending on the percentage of an installation that succeeds. If you specify a percent success for installations, Configuration Manager raises an alert in the console when the deployment succeeds on that percentage of installations. Use this parameter in conjunction with the *PercentSuccess* parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeploymentAvailableDay<DateTime>

Specifies a day, in MM/DD/YYYY format, when a deployment becomes available to clients. By default, the deployment becomes available immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeploymentAvailableTime<DateTime>

Specifies a time, in HH:MM format, when a deployment becomes available to clients. By default, the deployment becomes available immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeploymentExpireDay<DateTime>

Specifies a day, in MM/DD/YYYY format, when a deployment expires. By default, a deployment never expires. To expire a deployment on a certain day, set this parameter. You may use this parameter in conjunction with the *DeploymentExpireTime* parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-DeploymentExpireTime<DateTime>

Specifies a time, in HH:MM format, when the deployment expires. By default, a deployment never expires. To expire a deployment at a certain time, set this parameter. You may use this parameter in conjunction with the *DeploymentExpireDay* parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeploymentOption<DeploymentOptionType>

Specifies if clients download all content before starting the task sequence, or download content as needed by the running task sequence. By default, clients download content as needed. Valid values are:

- DownloadAllContentLocallyBeforeStartingTaskSequence
- DownloadContentLocallyWhenNeededByRunningTaskSequence

The acceptable values for this parameter are:

DownloadAllContentLocallyBeforeStartingTaskSequence	
DownloadContentLocallyWhenNeededByRunningTaskSequence	

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InternetOption<Boolean>

Indicates whether the task sequence runs on clients connecting over the Internet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-MakeAvailableTo<MakeAvailableToType>

Specifies whether to make this task sequence available to Configuration Manager clients, and whether to make it available when you deploy an operating system by using boot media, prestaged media, or PXE. Valid values are:

- Clients
- ClientsMediaAndPxe
- MediaAndPxe
- MediaAndPxeHidden

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PercentFailure<Int32>

Specifies a threshold percentage for failed task sequence deployment.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PercentSuccess<Int32>

Specifies a threshold percentage for successful task sequence deployment.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PersistOnWriteFilterDevice<Boolean>

Indicates whether to install a task sequence on the temporary overlay and commit changes later, or commit the changes at an installation deadline or a maintenance window.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RerunBehavior<RerunBehaviorType>

Specifies that a task sequence will be rerun on a computer if it has previously been run before the scheduled mandatory time. By default, the task sequence is always rerun. Valid values are:

-- AlwaysRerunProgram

-- NeverRerunDeployedProgram
 -- RerunIfFailedPreviousAttempt
 -- RerunIfSucceededOnPreviousAttempt
 The acceptable values for this parameter are:

AlwaysRerunProgram	
NeverRerunDeployedProgram	
RerunIfFailedPreviousAttempt	
RerunIfSucceededOnPreviousAttempt	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Schedule<IResultObject[]>

Specifies an array of **CMSchedule** objects. A **CMSchedule** object defines the mandatory assignment schedule for a deployment. To create a **CMSchedule** object, use the **New-CMSchedule** cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ScheduleEvent<ScheduleEventType[]>

Specifies an array of events that determine when the task sequence deployment runs. Valid values are:
 -- AsSoonAsPossible

-- LogOff

-- LogOn

The acceptable values for this parameter are:

AsSoonAsPossible	
LogOff	
LogOn	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SendWakeUpPacket<Boolean>

Indicates whether to send a wake up packet to computers before the deployment begins. If this value is \$True, Configuration Manager wakes a computer from sleep. If this value is \$False, it does not wake computers from sleep. For computers to wake, you must first configure Wake On LAN.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ShowTaskSequenceProgress<Boolean>

Indicates whether to show a process dialog for a task sequence.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SoftwareInstallation<Boolean>

Indicates whether to allow the application to install, even if the installation occurs outside of a maintenance window.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SystemRestart<Boolean>

Indicates whether to allow an advertised program to restart the system, even if the restart occurs outside of a maintenance window.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-TaskSequence<IResultObject>

Specifies a task sequence object. To obtain a task sequence object, use the **Get-CMTaskSequence** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-TaskSequenceDeploymentId<String>

Specifies an ID for a task sequence deployment.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-TaskSequenceName<String>

Specifies a name for a task sequence.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-TaskSequencePackageId<String>

Specifies an ID for a task sequence package.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UseMeteredNetwork<Boolean>

Indicates whether to allow clients on a metered Internet connection to download content after the installation deadline, which might incur additional costs.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UseUtcForAvailableSchedule<Boolean>

Indicates whether client computers use UTC time to determine the availability of a program. UTC time makes the task sequence available at the same time for all computers.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UseUtcForExpireSchedule<Boolean>

Indicates whether client computers use UTC time to determine the expiration of a program. UTC time makes the task sequence available at the same time for all computers.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Create a task sequence deployment

This command creates the task sequence deployment by using the task sequence name and collection name.

```
PS C:\> Set-CMTaskSequenceDeployment -TaskSequenceName "Task Sequence 1333" -CollectionName "All Systems" -Comment "Task sequence test" -ShowTaskSequenceProgress $True
```

Example 2: Create a task sequence deployment with a task sequence name

This command creates the task sequence deployment by using the task sequence name and collection name.

```
PS C:\> Set-CMTaskSequenceDeployment -TaskSequenceName "Task Sequence 1333" -CollectionName "All Desktop and Server Clients" -Comment "Task sequence test" -SendWakeUpPacket $True -UseMeteredNetwork $True -DeploymentExpireDay 2014/12/30 -DeploymentExpireTime 15:52 -UseUtcForExpireSchedule $True -ScheduleEvent LogOff -RerunBehavior NeverRerunDeployedProgram -AllowUsersRunIndependently $True -ShowTaskSequenceProgress $False -SoftwareInstallation $True -SystemRestart $True -PersistOnWriteFilterDevice $False -InternetOption $True -DeploymentOption DownloadAllContentLocallyBeforeStartingTaskSequence -AllowFallback $True -AllowSharedContent $True -CreatAlertBaseOnPercentSuccess $True -CreatAlertBaseOnPercentFailure $True
```

Related topics

[Start-CMTaskSequenceDeployment](#)

[New-CMSchedule](#)

[Get-CMTaskSequence](#)

Set-CMTrustedRootCertificateProfileConfigurationItem

Set-CMTrustedRootCertificateProfileConfigurationItem

Sets a root certificate profile.

Syntax

Parameter Set: SetByName

```
Set-CMTrustedRootCertificateProfileConfigurationItem -Name <String[]> [-DesiredConfigurationDigestPath <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetById

```
Set-CMTrustedRootCertificateProfileConfigurationItem -Id <String[]> [-DesiredConfigurationDigestPath <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetValue

```
Set-CMTrustedRootCertificateProfileConfigurationItem -InputObject <IResultObject> [-DesiredConfigurationDigestPath <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Set-CMTrustedRootCertificateProfileConfigurationItem** cmdlet sets a root certificate profile. Client computers use root certificate profiles to chain their certificates back to a corporate public key infrastructure (PKI) certification authority.

Parameters

-DesiredConfigurationDigestPath<String>

Specifies a path to the configuration data stored as a digest.

Aliases	none
Required?	false
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Id<String[]>

Specifies an array of IDs of root certificate profiles.

Aliases	CId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a root certificate profile object. To obtain a root certificate profile object use the **Get-
CMTrustedRootCertificateProfileConfigurationItem** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String[]>

Specifies an array of names of root certificate profiles.

Aliases	LocalizedDisplayName
Required?	true

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

[Copy-CMTrustedRootCertificateProfileConfigurationItem](#)

[Get-CMTrustedRootCertificateProfileConfigurationItem](#)

[New-CMTrustedRootCertificateProfileConfigurationItem](#)

[Remove-CMTrustedRootCertificateProfileConfigurationItem](#)

Set-CMUserCollection

Set-CMUserCollection

Sets the properties of one or more user collections in the Configuration Manager hierarchy.

Syntax

Parameter Set: SetByIdMandatory

```
Set-CMUserCollection -CollectionId <String> [-Comment <String> ] [-LimitingCollectionId <String> ] [-LimitingCollectionName <String> ] [-NewName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetByNameMandatory

```
Set-CMUserCollection -Name <String> [-Comment <String> ] [-LimitingCollectionId <String> ] [-LimitingCollectionName <String> ] [-NewName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetByValueMandatory

```
Set-CMUserCollection -InputObject <IResultObject> [-Comment <String> ] [-LimitingCollectionId <String> ] [-LimitingCollectionName <String> ] [-NewName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Set-CMUserCollection** cmdlet updates the name, description, or limiting collection of user collections.

Collections represent logical groupings of resources, such as users and devices. For more information about Microsoft System Center 2012 Configuration Manager collections, see [Introduction to Collections in Configuration Manager](http://go.microsoft.com/fwlink/p/?LinkID=259433) (<http://go.microsoft.com/fwlink/p/?LinkID=259433>) on TechNet.

Parameters

-CollectionId<String>

Specifies the IDs of the user collections that you want to update.

Aliases	none
Required?	true

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Comment<String>

Specifies a description of the specified user collections.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies an object that represents the user collections that you want to update.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-LimitingCollectionId<String>

Specifies the ID of the limiting collection for the specified user collections.

Aliases	LimitToCollectionId
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-LimitingCollectionName<String>

Specifies the name of the limiting collection for the specified user collections.

Aliases	LimitToCollectionName
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies the names of the user collections that you want to update.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-NewName<String>

Specifies a name for the specified user collections.

Aliases	none
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Modify a user collection

This command changes the name of the user collection named UC08 to UserCollection08. The command adds a comment for the collection and specifies a limiting collection.

```
PS C:\> Set-CMUserCollection -Name "UC08" -Comment "User collection 08." -  
LimitingCollectionName "All Users" -NewName "UserCollection08"
```

Related topics

[Export-CMUserCollection](#)

[Get-CMUserCollection](#)

[Import-CMUserCollection](#)

[New-CMUserCollection](#)

[Remove-CMUserCollection](#)

Set-CMUserDataAndProfileConfigurationItem

Set-CMUserDataAndProfileConfigurationItem

Modifies a user data and profile configuration item.

Syntax

Parameter Set: SetByName

```
Set-CMUserDataAndProfileConfigurationItem -Name <String[]> [-AccessPolicy <Boolean> ] [-
AddAdminGroupToRUPEnabled <Boolean> ] [-AllowAllDevice <Boolean> ] [-
AllowCrossForestUserPolicy <Boolean> ] [-BackgroundSynchronization {Disabled | Enabled |
NotConfigured} ] [-ConfigureFolderRedirection <Boolean> ] [-ConfigureOfflineFile <Boolean> ]
[-ConfigureRoamingUserProfile <Boolean> ] [-ConnectionTransferRate <Int32> ] [-
DeleteProfileOlderDays <Int32> ] [-DeleteRoamingCacheEnabled <Boolean> ] [-Description
<String> ] [-DetectSlowLink <Boolean> ] [-DeviceType
{FolderRedirectionOnAnyDeviceCachingOnPrimaryDevicesOnly | OnAnyDevice |
OnlyOnPrimaryDevices} ] [-DisableMakeOffline <Boolean> ] [-DisableWorkOffline <Boolean> ] [-
EnableOfflineFile <Boolean> ] [-EnableSlowLink <Boolean> ] [-ErrorDays <Int32> ] [-
ExcludeList <String[]> ] [-FileSynchronization {Disabled | Enabled | NotConfigured} ] [-
FolderRedirectionUserConfigurationForApplicationData {DoNotManage | RedirectToLocal |
RedirectToRemote} ] [-FolderRedirectionUserConfigurationForContacts {DoNotManage |
RedirectToLocal | RedirectToRemote} ] [-FolderRedirectionUserConfigurationForDesktop
{DoNotManage | RedirectToLocal | RedirectToRemote} ] [-
FolderRedirectionUserConfigurationForDocuments {DoNotManage | RedirectToLocal |
RedirectToRemote} ] [-FolderRedirectionUserConfigurationForDownloads {DoNotManage |
RedirectToLocal | RedirectToRemote} ] [-FolderRedirectionUserConfigurationForFavorites
{DoNotManage | RedirectToLocal | RedirectToRemote} ] [-
FolderRedirectionUserConfigurationForLinks {DoNotManage | RedirectToLocal |
RedirectToRemote} ] [-FolderRedirectionUserConfigurationForMusic {DoNotManage |
RedirectToLocal | RedirectToRemote} ] [-FolderRedirectionUserConfigurationForPictures
{DoNotManage | RedirectToLocal | RedirectToRemote} ] [-
FolderRedirectionUserConfigurationForSavedGames {DoNotManage | RedirectToLocal |
RedirectToRemote} ] [-FolderRedirectionUserConfigurationForSearches {DoNotManage |
RedirectToLocal | RedirectToRemote} ] [-FolderRedirectionUserConfigurationForStartMenu
{DoNotManage | RedirectToLocal | RedirectToRemote} ] [-
FolderRedirectionUserConfigurationForVideos {DoNotManage | RedirectToLocal |
RedirectToRemote} ] [-ForceUnloadDisabled <Boolean> ] [-GrantExclusiveRight <Boolean> ] [-
LeaveFolderNewLocation <Boolean> ] [-LimitDisk <Int32> ] [-ManageAdvancedSetting <Boolean> ]
[-ManageSlowLink <Boolean> ] [-MoveCachedFolder <Boolean> ] [-MoveContent <Boolean> ] [-
NewName <String> ] [-OfflineFile <String[]> ] [-OnlyAllowLocalProfiles <Boolean> ] [-
OwnerCheckDisabled <Boolean> ] [-ProfileUploadDisabled <Boolean> ] [-SlowLink <Int32> ] [-
SlowLinkUIEnabled <Boolean> ] [-SpecifiedLocation <String> ] [-SpecifyTime <String> ] [-
SpecifyTimeInterval <Int32> ] [-SynchronizationInterval <Int32> ] [-SynchronizationList
<String[]> ] [-SynchronizationPolicy <Boolean> ] [-TempProfileLogonBlocked <Boolean> ] [-
Timeout <Int32> ] [-UseCommonAlert <Boolean> ] [-UseSpecifiedLocation <Boolean> ] [-
WaitForNetworkInSeconds <Int32> ] [-WarningDays <Int32> ] [-Confirm] [-WhatIf] [
<CommonParameters>]
```

Parameter Set: SetById

```
Set-CMUserDataAndProfileConfigurationItem -Id <String[]> [-AccessPolicy <Boolean> ] [-
AddAdminGroupToRUPEnabled <Boolean> ] [-AllowAllDevice <Boolean> ] [-
AllowCrossForestUserPolicy <Boolean> ] [-BackgroundSynchronization {Disabled | Enabled |
NotConfigured} ] [-ConfigureFolderRedirection <Boolean> ] [-ConfigureOfflineFile <Boolean> ]
[-ConfigureRoamingUserProfile <Boolean> ] [-ConnectionTransferRate <Int32> ] [-
DeleteProfileOlderDays <Int32> ] [-DeleteRoamingCacheEnabled <Boolean> ] [-Description
<String> ] [-DetectSlowLink <Boolean> ] [-DeviceType
{FolderRedirectionOnAnyDeviceCachingOnPrimaryDevicesOnly | OnAnyDevice |
OnlyOnPrimaryDevices} ] [-DisableMakeOffline <Boolean> ] [-DisableWorkOffline <Boolean> ] [-
EnableOfflineFile <Boolean> ] [-EnableSlowLink <Boolean> ] [-ErrorDays <Int32> ] [-
ExcludeList <String[]> ] [-FileSynchronization {Disabled | Enabled | NotConfigured} ] [-
FolderRedirectionUserConfigurationForApplicationData {DoNotManage | RedirectToLocal |
RedirectToRemote} ] [-FolderRedirectionUserConfigurationForContacts {DoNotManage |
RedirectToLocal | RedirectToRemote} ] [-FolderRedirectionUserConfigurationForDesktop
{DoNotManage | RedirectToLocal | RedirectToRemote} ] [-
FolderRedirectionUserConfigurationForDocuments {DoNotManage | RedirectToLocal |
RedirectToRemote} ] [-FolderRedirectionUserConfigurationForDownloads {DoNotManage |
RedirectToLocal | RedirectToRemote} ] [-FolderRedirectionUserConfigurationForFavorites
{DoNotManage | RedirectToLocal | RedirectToRemote} ] [-
FolderRedirectionUserConfigurationForLinks {DoNotManage | RedirectToLocal |
RedirectToRemote} ] [-FolderRedirectionUserConfigurationForMusic {DoNotManage |
RedirectToLocal | RedirectToRemote} ] [-FolderRedirectionUserConfigurationForPictures
{DoNotManage | RedirectToLocal | RedirectToRemote} ] [-
FolderRedirectionUserConfigurationForSavedGames {DoNotManage | RedirectToLocal |
RedirectToRemote} ] [-FolderRedirectionUserConfigurationForSearches {DoNotManage |
RedirectToLocal | RedirectToRemote} ] [-FolderRedirectionUserConfigurationForStartMenu
{DoNotManage | RedirectToLocal | RedirectToRemote} ] [-
FolderRedirectionUserConfigurationForVideos {DoNotManage | RedirectToLocal |
RedirectToRemote} ] [-ForceUnloadDisabled <Boolean> ] [-GrantExclusiveRight <Boolean> ] [-
LeaveFolderNewLocation <Boolean> ] [-LimitDisk <Int32> ] [-ManageAdvancedSetting <Boolean> ]
[-ManageSlowLink <Boolean> ] [-MoveCachedFolder <Boolean> ] [-MoveContent <Boolean> ] [-
NewName <String> ] [-OfflineFile <String[]> ] [-OnlyAllowLocalProfiles <Boolean> ] [-
OwnerCheckDisabled <Boolean> ] [-ProfileUploadDisabled <Boolean> ] [-SlowLink <Int32> ] [-
SlowLinkUIEnabled <Boolean> ] [-SpecifiedLocation <String> ] [-SpecifyTime <String> ] [-
SpecifyTimeInterval <Int32> ] [-SynchronizationInterval <Int32> ] [-SynchronizationList
<String[]> ] [-SynchronizationPolicy <Boolean> ] [-TempProfileLogonBlocked <Boolean> ] [-
TimeOut <Int32> ] [-UseCommonAlert <Boolean> ] [-UseSpecifiedLocation <Boolean> ] [-
WaitForNetworkInSeconds <Int32> ] [-WarningDays <Int32> ] [-Confirm] [-WhatIf] [
<CommonParameters>]
```

Parameter Set: SetByValue

```
Set-CMUserDataAndProfileConfigurationItem -InputObject <IResultObject> [-AccessPolicy
<Boolean> ] [-AddAdminGroupToRUPEnabled <Boolean> ] [-AllowAllDevice <Boolean> ] [-
AllowCrossForestUserPolicy <Boolean> ] [-BackgroundSynchronization {Disabled | Enabled |
NotConfigured} ] [-ConfigureFolderRedirection <Boolean> ] [-ConfigureOfflineFile <Boolean> ]
[-ConfigureRoamingUserProfile <Boolean> ] [-ConnectionTransferRate <Int32> ] [-
DeleteProfileOlderDays <Int32> ] [-DeleteRoamingCacheEnabled <Boolean> ] [-Description
<String> ] [-DetectSlowLink <Boolean> ] [-DeviceType
{FolderRedirectionOnAnyDeviceCachingOnPrimaryDevicesOnly | OnAnyDevice |
OnlyOnPrimaryDevices} ] [-DisableMakeOffline <Boolean> ] [-DisableWorkOffline <Boolean> ] [-
```

```

EnableOfflineFile <Boolean> ] [-EnableSlowLink <Boolean> ] [-ErrorDays <Int32> ] [-
ExcludeList <String[]> ] [-FileSynchronization {Disabled | Enabled | NotConfigured} ] [-
FolderRedirectionUserConfigurationForApplicationData {DoNotManage | RedirectToLocal |
RedirectToRemote} ] [-FolderRedirectionUserConfigurationForContacts {DoNotManage |
RedirectToLocal | RedirectToRemote} ] [-FolderRedirectionUserConfigurationForDesktop
{DoNotManage | RedirectToLocal | RedirectToRemote} ] [-
FolderRedirectionUserConfigurationForDocuments {DoNotManage | RedirectToLocal |
RedirectToRemote} ] [-FolderRedirectionUserConfigurationForDownloads {DoNotManage |
RedirectToLocal | RedirectToRemote} ] [-FolderRedirectionUserConfigurationForFavorites
{DoNotManage | RedirectToLocal | RedirectToRemote} ] [-
FolderRedirectionUserConfigurationForLinks {DoNotManage | RedirectToLocal |
RedirectToRemote} ] [-FolderRedirectionUserConfigurationForMusic {DoNotManage |
RedirectToLocal | RedirectToRemote} ] [-FolderRedirectionUserConfigurationForPictures
{DoNotManage | RedirectToLocal | RedirectToRemote} ] [-
FolderRedirectionUserConfigurationForSavedGames {DoNotManage | RedirectToLocal |
RedirectToRemote} ] [-FolderRedirectionUserConfigurationForSearches {DoNotManage |
RedirectToLocal | RedirectToRemote} ] [-FolderRedirectionUserConfigurationForStartMenu
{DoNotManage | RedirectToLocal | RedirectToRemote} ] [-
FolderRedirectionUserConfigurationForVideos {DoNotManage | RedirectToLocal |
RedirectToRemote} ] [-ForceUnloadDisabled <Boolean> ] [-GrantExclusiveRight <Boolean> ] [-
LeaveFolderNewLocation <Boolean> ] [-LimitDisk <Int32> ] [-ManageAdvancedSetting <Boolean> ]
[-ManageSlowLink <Boolean> ] [-MoveCachedFolder <Boolean> ] [-MoveContent <Boolean> ] [-
NewName <String> ] [-OfflineFile <String[]> ] [-OnlyAllowLocalProfiles <Boolean> ] [-
OwnerCheckDisabled <Boolean> ] [-ProfileUploadDisabled <Boolean> ] [-SlowLink <Int32> ] [-
SlowLinkUIEnabled <Boolean> ] [-SpecifiedLocation <String> ] [-SpecifyTime <String> ] [-
SpecifyTimeInterval <Int32> ] [-SynchronizationInterval <Int32> ] [-SynchronizationList
<String[]> ] [-SynchronizationPolicy <Boolean> ] [-TempProfileLogonBlocked <Boolean> ] [-
TimeOut <Int32> ] [-UseCommonAlert <Boolean> ] [-UseSpecifiedLocation <Boolean> ] [-
WaitForNetworkInSeconds <Int32> ] [-WarningDays <Int32> ] [-Confirm] [-WhatIf] [
<CommonParameters>]

```

Detailed Description

The **Set-CMUserDataAndProfileConfigurationItem** cmdlet modifies a user data and profile configuration item that can apply to Windows 8 computers. A configuration item can manage folder redirection, offline folders, and roaming user profiles. You can create a configuration item by using the **New-CMUserDataAndProfileConfigurationItem** cmdlet.

Parameters

-AccessPolicy<Boolean>

Indicates whether this configuration item manages profile access settings for roaming profiles.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AddAdminGroupToRUPEnabled<Boolean>

Indicates whether to grant the Administrators group access to roaming profile folders.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AllowAllDevice<Boolean>

Indicates whether to allow roaming profiles on all devices. If this value is \$False, roaming profiles apply only to the primary device for a user.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AllowCrossForestUserPolicy<Boolean>

Indicates whether to permit user policies to roam across Active Directory forests that have a trust relationship with the current forest.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-BackgroundSynchronization<SynchronizationType>

Specifies a background synchronization type for file in offline mode. Valid values are: Disabled, Enabled, and NotConfigured.

The acceptable values for this parameter are:

Disabled	
Enabled	
NotConfigured	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ConfigureFolderRedirection<Boolean>

Indicates whether the configuration item includes settings for folder redirection.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ConfigureOfflineFile<Boolean>

Indicates whether the configuration item includes settings for offline folders.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ConfigureRoamingUserProfile<Boolean>

Indicates whether the configuration item includes settings for roaming user profiles.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ConnectionTransferRate<Int32>

Specifies a connection transfer rate.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeleteProfileOlderDays<Int32>

Specifies the number of days to keep a user profile since the last time someone used it. A computer deletes an older profile when it restarts.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeleteRoamingCacheEnabled<Boolean>

Indicates whether to delete cached copies of roaming user profiles. The default for this parameter is \$False.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Description<String>

Specifies a description for the configuration item.

Aliases	none
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DetectSlowLink<Boolean>

Indicates whether to detect slow links.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeviceType<DeviceType>

Specifies the applicability of folder redirection for user devices. Valid values are:

- FolderRedirectionOnAnyDeviceCachingOnPrimaryDevicesOnly. Folder redirection for any user device, but caching only on the primary device for a user.
- OnAnyDevice. Folder redirection and caching on any device.
- OnlyOnPrimaryDevices. Folder redirection and caching on the primary device for a user.

The acceptable values for this parameter are:

FolderRedirectionOnAnyDeviceCachingOnPrimaryDevicesOnly	
OnAnyDevice	
OnlyOnPrimaryDevices	

Aliases	none
Required?	false
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DisableMakeOffline<Boolean>

Indicates whether users can disable the **Make Available Offline** command.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DisableWorkOffline<Boolean>

Indicates whether users can disable the **Work Offline** command.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnableOfflineFile<Boolean>

Indicates whether this configuration item enables use of offline files.

Aliases	none
Required?	false
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnableSlowLink<Boolean>

Indicates whether the configuration enables work with offline files over a slow link.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ErrorDays<Int32>

Specifies the number of days to wait before the profile creates an error.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ExcludeList<String[]>

Specifies an array of folders. The configuration item excludes these folders from roaming profiles.

Aliases	none
Required?	false
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-FileSynchronization<SynchronizationType>

Specifies a file synchronization type for metered networks for work in offline mode. Valid values are: Disabled, Enabled, and NotConfigured.

The acceptable values for this parameter are:

Disabled	
Enabled	
NotConfigured	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-

FolderRedirectionUserConfigurationForApplicationData<FolderRedirectionType>

Specifies whether to redirect the Application Data folder to a local folder or a remote folder, or not to redirect this folder. Valid values are:

- DoNotManage
- RedirectToLocal
- RedirectToRemote

The acceptable values for this parameter are:

DoNotManage	
-------------	--

RedirectToLocal	
RedirectToRemote	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-

FolderRedirectionUserConfigurationForContacts<FolderRedirection Type>

Specifies whether to redirect the Contacts folder to a local folder or a remote folder, or not to redirect this folder. Valid values are:

- DoNotManage
- RedirectToLocal
- RedirectToRemote

The acceptable values for this parameter are:

DoNotManage	
RedirectToLocal	
RedirectToRemote	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-

FolderRedirectionUserConfigurationForDesktop<FolderRedirectionType>

Specifies whether to redirect the Desktop to a local folder or a remote folder, or not to redirect this folder. Valid values are:

- DoNotManage
- RedirectToLocal
- RedirectToRemote

The acceptable values for this parameter are:

DoNotManage	
RedirectToLocal	
RedirectToRemote	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-

FolderRedirectionUserConfigurationForDocuments<FolderRedirectionType>

Specifies whether to redirect the Documents folder to a local folder or a remote folder, or not to redirect this folder. Valid values are:

- DoNotManage
- RedirectToLocal
- RedirectToRemote

The acceptable values for this parameter are:

DoNotManage	
RedirectToLocal	
RedirectToRemote	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-

FolderRedirectionUserConfigurationForDownloads<FolderRedirectionType>

Specifies whether to redirect the Downloads folder to a local folder or a remote folder, or not to redirect this folder. Valid values are:

- DoNotManage
- RedirectToLocal
- RedirectToRemote

The acceptable values for this parameter are:

DoNotManage	
RedirectToLocal	
RedirectToRemote	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-

FolderRedirectionUserConfigurationForFavorites<FolderRedirectionType>

Specifies whether to redirect the Favorites folder to a local folder or a remote folder, or not to redirect this folder. Valid values are:

- DoNotManage
- RedirectToLocal
- RedirectToRemote

The acceptable values for this parameter are:

DoNotManage	
RedirectToLocal	
RedirectToRemote	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-

FolderRedirectionUserConfigurationForLinks<FolderRedirectionType>

Specifies whether to redirect the Links folder to a local folder or a remote folder, or not to redirect this folder. Valid values are:

- DoNotManage
- RedirectToLocal
- RedirectToRemote

The acceptable values for this parameter are:

DoNotManage	
RedirectToLocal	
RedirectToRemote	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-

FolderRedirectionUserConfigurationForMusic<FolderRedirectionType>

Specifies whether to redirect the Music folder to a local folder or a remote folder, or not to redirect this folder. Valid values are:

- DoNotManage
- RedirectToLocal
- RedirectToRemote

The acceptable values for this parameter are:

DoNotManage	
RedirectToLocal	
RedirectToRemote	

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-

FolderRedirectionUserConfigurationForPictures<FolderRedirectionType>

Specifies whether to redirect the Pictures folder to a local folder or a remote folder, or not to redirect this folder. Valid values are:

- DoNotManage
- RedirectToLocal
- RedirectToRemote

The acceptable values for this parameter are:

DoNotManage	
RedirectToLocal	
RedirectToRemote	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-

FolderRedirectionUserConfigurationForSavedGames<FolderRedirectionType>

Specifies whether to redirect the Saved Games folder to a local folder or a remote folder, or not to redirect this folder. Valid values are:

- DoNotManage
- RedirectToLocal
- RedirectToRemote

The acceptable values for this parameter are:

DoNotManage	
RedirectToLocal	
RedirectToRemote	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-

FolderRedirectionUserConfigurationForSearches<FolderRedirection Type>

Specifies whether to redirect the Searches folder to a local folder or a remote folder, or not to redirect this folder. Valid values are:

- DoNotManage
- RedirectToLocal
- RedirectToRemote

The acceptable values for this parameter are:

DoNotManage	
RedirectToLocal	
RedirectToRemote	

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-

FolderRedirectionUserConfigurationForStartMenu<FolderRedirectionType>

Specifies whether to redirect the Start Menu to a local folder or a remote folder, or not to redirect this folder. Valid values are:

- DoNotManage
- RedirectToLocal
- RedirectToRemote

The acceptable values for this parameter are:

DoNotManage	
RedirectToLocal	
RedirectToRemote	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-

FolderRedirectionUserConfigurationForVideos<FolderRedirectionType>

Specifies whether to redirect the Videos folder to a local folder or a remote folder, or not to redirect this folder. Valid values are:

- DoNotManage
- RedirectToLocal
- RedirectToRemote

The acceptable values for this parameter are:

DoNotManage	
RedirectToLocal	
RedirectToRemote	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ForceUnloadDisabled<Boolean>

Indicates whether to disable forced unload of a user profile at logoff. The default value for this parameter is \$False.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-GrantExclusiveRight<Boolean>

Indicates whether to grant the user exclusive permissions to a redirected folder.

Aliases	none
Required?	false
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Id<String[]>

Specifies an array of IDs for user data and profile configuration items.

Aliases	CId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a user data and profile configuration item object. To obtain a configuration item object, use the **Get-CMUserDataAndProfileConfigurationItem** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-LeaveFolderNewLocation<Boolean>

Indicates whether to leave the folder in the redirected location in the event you remove this configuration item.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-LimitDisk<Int32>

Specifies a limit, in megabytes, for the disk space used for offline files.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ManageAdvancedSetting<Boolean>

Indicates whether this configuration item manages advanced settings for folder redirection. Specify values for any of the following parameters:

- *GrantExclusiveRight*
- *MoveContent*
- *LeaveFolderNewLocation*
- *MoveCachedFolder*

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ManageSlowLink<Boolean>

Indicates whether this profile item manages slow links.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-MoveCachedFolder<Boolean>

Indicates whether to move the cached folder when the path updates on the server.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-MoveContent<Boolean>

Indicates whether to move the contents of redirected folders to the new location.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String[]>

Specifies an array of names of user data and profile configuration items.

Aliases	LocalizedDisplayName
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-NewName<String>

Specifies a new name for the configuration item.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OfflineFile<String[]>

Specifies an array of Administrative user assigned offline folders, as UNC paths as follows:

\\server\share\%UserName%.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OnlyAllowLocalProfiles<Boolean>

Indicates whether the configuration item allows only local profiles, not domain profiles.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OwnerCheckDisabled<Boolean>

Indicates whether the configuration item does not check for ownership of roaming profile folders.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ProfileUploadDisabled<Boolean>

Indicates whether to disable uploading of profiles.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SlowLink<Int32>

Specifies a value for a slow link.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SlowLinkUIEnabled<Boolean>

Indicates whether to enable user logon prompt to allow profile download when a device detects a slow link.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SpecifiedLocation<String>

Specifies a specified location.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SpecifyTime<String>

Specifies a time for background upload of the user hive.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SpecifyTimeInterval<Int32>

Specifies a time of day for synchronization for user profiles. Enter a value of zero (0) to 12. Zero indicates 12:00 a.m. Values from one (1) to 12 indicate times from 1:00 p.m. to 12:00 p.m.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SynchronizationInterval<Int32>

Specifies a synchronization interval, in hours, for the user profiles.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SynchronizationList<String[]>

Specifies an array of folders. The configuration item specifies these subfolders of Appdata\Roaming to synchronize only at logon and logoff.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SynchronizationPolicy<Boolean>

Indicates whether to use a synchronization policy.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-TempProfileLogonBlocked<Boolean>

Indicates whether to block users from logging on by using a temporary profile.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Timeout<Int32>

Specifies a timeout value.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UseCommonAlert<Boolean>

Indicates whether to use common alerts.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UseSpecifiedLocation<Boolean>

Indicates whether to use the specified location referred to by the *SpecifiedLocation* parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-WaitForNetworkInSeconds<Int32>

Specifies the maximum time to wait, in seconds, for slow link network connectivity before loading the profile.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-WarningDays<Int32>

Specifies the number of days to wait before the profile creates a warning.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Modify a configuration item

This command modifies the configuration item named CMUDaPCI001, changing its name to CMUDaPCI0073. The command also sets values for whether the configuration item specifies folder redirection, offline folders, and roaming profiles, and it provides a description for the configuration item.

```
PS C:\> Set-CMUserDataAndProfileConfigurationItem -Name "CMUDaPCI001" -NewName
"CMUDaPCI0073" -ConfigureFolderRedirection $False -ConfigureOfflineFile $False -
ConfigureRoamingUserProfile $False -Description "User data and profile configuration
information"
```

Related topics

[Copy-CMUserDataAndProfileConfigurationItem](#)

[Get-CMUserDataAndProfileConfigurationItem](#)

[New-CMUserDataAndProfileConfigurationItem](#)

[Remove-CMUserDataAndProfileConfigurationItem](#)

Set-CMVhd

Set-CMVhd

Modifies VHD images.

Syntax

Parameter Set: SetById

```
Set-CMVhd -PackageID <String[]> [-Description <String> ] [-DistributionPointServerNames <String[]> ] [-NewName <String> ] [-Path <String> ] [-SecurityScopeAction {AddMembership | RemoveMembership} ] [-SecurityScopeName <String> ] [-TaskSequencePackageId <String> ] [-Version <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetByName

```
Set-CMVhd -Name <String[]> [-Description <String> ] [-DistributionPointServerNames <String[]> ] [-NewName <String> ] [-Path <String> ] [-SecurityScopeAction {AddMembership | RemoveMembership} ] [-SecurityScopeName <String> ] [-TaskSequencePackageId <String> ] [-Version <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetByValue

```
Set-CMVhd -InputObject <IResultObject> [-Description <String> ] [-DistributionPointServerNames <String[]> ] [-NewName <String> ] [-Path <String> ] [-SecurityScopeAction {AddMembership | RemoveMembership} ] [-SecurityScopeName <String> ] [-TaskSequencePackageId <String> ] [-Version <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetSecurityScopeById

```
Set-CMVhd -SecuredScopeNames <String[]> [-Description <String> ] [-DistributionPointServerNames <String[]> ] [-NewName <String> ] [-PackageID <String[]> ] [-Path <String> ] [-SecurityScopeAction {AddMembership | RemoveMembership} ] [-SecurityScopeName <String> ] [-TaskSequencePackageId <String> ] [-Version <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetSecurityScopeByName

```
Set-CMVhd -SecuredScopeNames <String[]> [-Description <String> ] [-DistributionPointServerNames <String[]> ] [-Name <String[]> ] [-NewName <String> ] [-Path <String> ] [-SecurityScopeAction {AddMembership | RemoveMembership} ] [-SecurityScopeName <String> ] [-TaskSequencePackageId <String> ] [-Version <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Set-CMVhd** cmdlet modifies one or more virtual hard disk (VHD) images that were created through the operating system deployment feature.

Parameters

-Description<String>

Specifies a description for the VHD.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DistributionPointServerNames<String[]>

Specifies an array of names of distribution point servers, which contain content that the task sequence requires.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a VHD image. To obtain a VHD image, use the **Get-CMVhd** cmdlet.

Aliases	none
Required?	true

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue, ByPropertyName)
Accept Wildcard Characters?	false

-Name<String[]>

Specifies an array of names of VHD images.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-NewName<String>

Specifies a new name for the VHD image.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PackageID<String[]>

Specifies an array of IDs of packages.

Aliases	Id
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Path<String>

Specifies a path for the VHD image.

Aliases	PackageSourcePath
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SecuredScopeNames<String[]>

Specifies an array of names of security scopes. A security scope name can be Default or the name of a custom-created security scope.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SecurityScopeAction<SecurityScopeActionType>

Specifies an action for a security scope. Valid values are: AddMembership and RemoveMembership.

The acceptable values for this parameter are:

AddMembership	
RemoveMembership	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SecurityScopeName<String>

Specifies the name of a security scope. A security scope name can be Default or the name of a custom security scope.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-TaskSequencePackageId<String>

Specifies an ID for a task sequence package.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Version<String>

Specifies a version for the VHD. Use any string.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

[Get-CMVhd](#)

[New-CMVhd](#)

[Remove-CMVhd](#)

Set-CMVpnProfileConfigurationItem

Set-CMVpnProfileConfigurationItem

Modifies a VPN profile.

Syntax

Parameter Set: SetByName

```
Set-CMVpnProfileConfigurationItem -Name <String[]> [-DesiredConfigurationDigestPath <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetById

```
Set-CMVpnProfileConfigurationItem -Id <String[]> [-DesiredConfigurationDigestPath <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetByValue

```
Set-CMVpnProfileConfigurationItem -InputObject <IResultObject> [-DesiredConfigurationDigestPath <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Set-CMVpnProfileConfigurationItem** cmdlet modifies a virtual private network (VPN) profile. Client computers use VPN profiles to remotely connect to a company network over the Internet.

Parameters

-DesiredConfigurationDigestPath<String>

Specifies a path to the configuration data stored as a digest.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Id<String[]>

Specifies an array of IDs of VPN profile objects.

Aliases	CId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a VPN profile object. To obtain a VPN profile object, use the **Get-CMVpnProfileConfigurationItem** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String[]>

Specifies an array of names of VPN profiles.

Aliases	LocalizedDisplayName
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

[Copy-CMVpnProfileConfigurationItem](#)

[Get-CMVpnProfileConfigurationItem](#)

[New-CMVpnProfileConfigurationItem](#)

[Remove-CMVpnProfileConfigurationItem](#)

Set-CMWindowsFirewallPolicy

Set-CMWindowsFirewallPolicy

Changes settings of a Windows Firewall policy.

Syntax

Parameter Set: SearchByNameMandatory

```
Set-CMWindowsFirewallPolicy -Name <String[]> [-Description <String> ] [-DomainBlockAllInboundTraffic {No | NotConfigured | Yes} ] [-DomainNotifications {No | NotConfigured | Yes} ] [-DomainTurnOnFirewall {No | NotConfigured | Yes} ] [-NewName <String> ] [-PrivateBlockAllInboundTraffic {No | NotConfigured | Yes} ] [-PrivateNotifications {No | NotConfigured | Yes} ] [-PrivateTurnOnFirewall {No | NotConfigured | Yes} ] [-PublicBlockAllInboundTraffic {No | NotConfigured | Yes} ] [-PublicNotifications {No | NotConfigured | Yes} ] [-PublicTurnOnFirewall {No | NotConfigured | Yes} ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Set-CMWindowsFirewallPolicy -Id <String[]> [-Description <String> ] [-DomainBlockAllInboundTraffic {No | NotConfigured | Yes} ] [-DomainNotifications {No | NotConfigured | Yes} ] [-DomainTurnOnFirewall {No | NotConfigured | Yes} ] [-NewName <String> ] [-PrivateBlockAllInboundTraffic {No | NotConfigured | Yes} ] [-PrivateNotifications {No | NotConfigured | Yes} ] [-PrivateTurnOnFirewall {No | NotConfigured | Yes} ] [-PublicBlockAllInboundTraffic {No | NotConfigured | Yes} ] [-PublicNotifications {No | NotConfigured | Yes} ] [-PublicTurnOnFirewall {No | NotConfigured | Yes} ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Set-CMWindowsFirewallPolicy -InputObject <IResultObject> [-Description <String> ] [-DomainBlockAllInboundTraffic {No | NotConfigured | Yes} ] [-DomainNotifications {No | NotConfigured | Yes} ] [-DomainTurnOnFirewall {No | NotConfigured | Yes} ] [-NewName <String> ] [-PrivateBlockAllInboundTraffic {No | NotConfigured | Yes} ] [-PrivateNotifications {No | NotConfigured | Yes} ] [-PrivateTurnOnFirewall {No | NotConfigured | Yes} ] [-PublicBlockAllInboundTraffic {No | NotConfigured | Yes} ] [-PublicNotifications {No | NotConfigured | Yes} ] [-PublicTurnOnFirewall {No | NotConfigured | Yes} ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetById

```
Set-CMWindowsFirewallPolicy -Id <String[]> -Priority {Decrease | Increase} [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetByName

```
Set-CMWindowsFirewallPolicy -Name <String[]> -Priority {Decrease | Increase} [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetByValue

```
Set-CMWindowsFirewallPolicy -InputObject <IResultObject> -Priority {Decrease | Increase} [-
```

Confirm] [-WhatIf] [<CommonParameters>]

Detailed Description

The **Set-CMWindowsFirewallPolicy** cmdlet changes settings of one or more Windows Firewall policies for System Center 2012 Endpoint Protection in Microsoft System Center 2012 Configuration Manager.

Parameters

-Description<String>

Specifies a description for the Windows Firewall policy.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DomainBlockAllInboundTraffic<SettingType>

Specifies whether the firewall blocks all incoming traffic for a domain type network location. Valid values are:

- No
- Not Configured
- Yes

The acceptable values for this parameter are:

No	
NotConfigured	
Yes	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DomainNotifications<SettingType>

Specifies whether Configuration Manager sends notifications to domain type network locations. Valid values are:

- No
- Not Configured
- Yes

The acceptable values for this parameter are:

No	
NotConfigured	
Yes	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DomainTurnOnFirewall<SettingType>

Specifies whether to enable Windows Firewall for domain network location. Valid values are:

- No
- Not Configured
- Yes

The acceptable values for this parameter are:

No	
NotConfigured	
Yes	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Id<String[]>

Specifies an array of IDs of firewall policies.

Aliases	CIId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a **CMWindowsFirewallPolicy** object. To obtain a **CMWindowsFirewallPolicy** object, use the **Get-CMWindowsFirewallPolicy** cmdlet.

Aliases	none
Required?	true
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String[]>

Specifies an array of firewall policy names.

Aliases	LocalizedDisplayName
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-NewName<String>

Specifies a new name for the firewall policy.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Priority<PriorityChangeType>

Specifies the priority of a firewall policy. Valid values are: Increase and Decrease.

The acceptable values for this parameter are:

Decrease	
Increase	

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PrivateBlockAllInboundTraffic<SettingType>

Specifies whether the firewall blocks all incoming traffic for a private network location. Valid values are:

- No
- Not Configured
- Yes

The acceptable values for this parameter are:

No	
NotConfigured	
Yes	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PrivateNotifications<SettingType>

Specifies whether Configuration Manager sends notifications to private network locations. Valid values are:

- No

-- Not Configured

-- Yes

The acceptable values for this parameter are:

No	
NotConfigured	
Yes	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PrivateTurnOnFirewall<SettingType>

Specifies whether to enable Windows Firewall for a private network location. Valid values are:

-- No

-- Not Configured

-- Yes

The acceptable values for this parameter are:

No	
NotConfigured	
Yes	

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PublicBlockAllInboundTraffic<SettingType>

Specifies whether the firewall blocks all incoming traffic for a public network location. Valid values are:

- No
- Not Configured
- Yes

The acceptable values for this parameter are:

No	
NotConfigured	
Yes	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PublicNotifications<SettingType>

Specifies whether Configuration Manager sends notifications to public network locations. Valid values are:

- No
- Not Configured
- Yes

The acceptable values for this parameter are:

No	
NotConfigured	

Yes	
-----	--

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PublicTurnOnFirewall<SettingType>

Specifies whether to enable Windows Firewall for a public network location. Valid values are:

- No
- Not Configured
- Yes

The acceptable values for this parameter are:

No	
NotConfigured	
Yes	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Decrease the priority of a Windows Firewall policy by using a name

This command decreases the priority of the Windows Firewall policy named WFPContos01.

```
PS C:\> Set-CMWindowsFirewallPolicy -Priority Decrease -Name "WFPContos01"
```

Example 2: Decrease the priority of a Windows Firewall policy by using an ID

This command decreases the priority of the Windows Firewall policy that has the ID 16777568.

```
PS C:\> Set-CMWindowsFirewallPolicy -Priority Decrease -Id "16777568"
```

Example 3: Increase the priority of a Windows Firewall policy by using an object variable

The first command gets the **CMWindowsFirewallPolicy** object that has the ID 16777568 and stores it in the \$WFPobj variable.

The second command increases the priority of the Windows Firewall policy stored in the \$WFPobj variable.

```
PS C:\> $WFPobj=Get-CMWindowsFirewallPolicy -Id "16777568"
```

```
PS C:\> Set-CMWindowsFirewallPolicy -Priority Increase -InputObject $WFPobj
```

Related topics

[New-CMWindowsFirewallPolicy](#)

[Get-CMWindowsFirewallPolicy](#)

[Remove-CMWindowsFirewallPolicy](#)

Set-CMWirelessProfileConfigurationItem

Set-CMWirelessProfileConfigurationItem

Modifies a wireless profile.

Syntax

Parameter Set: SetByName

```
Set-CMWirelessProfileConfigurationItem -Name <String[]> [-DesiredConfigurationDigestPath <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetById

```
Set-CMWirelessProfileConfigurationItem -Id <String[]> [-DesiredConfigurationDigestPath <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SetValue

```
Set-CMWirelessProfileConfigurationItem -InputObject <IResultObject> [-DesiredConfigurationDigestPath <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Set-CMWirelessProfileConfigurationItem** cmdlet modifies one or more wireless profiles. Client computers use wireless profiles for configuration when they connect to a company wireless network.

Parameters

-DesiredConfigurationDigestPath<String>

Specifies a path to the configuration data stored as a digest.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Id<String[]>

Specifies an array of IDs of wireless profiles.

Aliases	CId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a wireless profile object. To obtain a wireless profile object, use the **Get-CMWirelessProfileConfigurationItem** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String[]>

Specifies an array of names of wireless profiles.

Aliases	LocalizedDisplayName
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

[Copy-CMWirelessProfileConfigurationItem](#)

[Get-CMWirelessProfileConfigurationItem](#)

[New-CMWirelessProfileConfigurationItem](#)

[Remove-CMWirelessProfileConfigurationItem](#)

Start-CMAntimalwarePolicyDeployment

Start-CMAntimalwarePolicyDeployment

Starts the deployment of an antimalware policy to the members of a Configuration Manager collection.

Syntax

Parameter Set: SearchByAntimalwarePolicyId_CollectionId

```
Start-CMAntimalwarePolicyDeployment -AntimalwarePolicyId <String> -CollectionId <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByAntimalwarePolicyId_CollectionName

```
Start-CMAntimalwarePolicyDeployment -AntimalwarePolicyId <String> -CollectionName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByAntimalwarePolicyId_CollectionValue

```
Start-CMAntimalwarePolicyDeployment -AntimalwarePolicyId <String> -Collection <IResultObject> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByAntimalwarePolicyName_CollectionId

```
Start-CMAntimalwarePolicyDeployment -AntimalwarePolicyName <String> -CollectionId <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByAntimalwarePolicyName_CollectionName

```
Start-CMAntimalwarePolicyDeployment -AntimalwarePolicyName <String> -CollectionName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByAntimalwarePolicyName_CollectionValue

```
Start-CMAntimalwarePolicyDeployment -AntimalwarePolicyName <String> -Collection <IResultObject> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByAntimalwarePolicyValue_CollectionId

```
Start-CMAntimalwarePolicyDeployment -AntimalwarePolicy <IResultObject> -CollectionId <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByAntimalwarePolicyValue_CollectionName

```
Start-CMAntimalwarePolicyDeployment -AntimalwarePolicy <IResultObject> -CollectionName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByAntimalwarePolicyValue_CollectionValue

```
Start-CMAntimalwarePolicyDeployment -AntimalwarePolicy <IResultObject> -Collection <IResultObject> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: StartDeploymentByName

```
Start-CMAntimalwarePolicyDeployment -Name <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Start-CMAntimalwarePolicyDeployment** cmdlet starts the deployment of a Microsoft System Center 2012 Endpoint Protection antimalware policy to the members of a Microsoft System Center 2012 Configuration Manager collection.

To start a policy deployment, specify the antimalware policy to be deployed and the device collection that is the destination for the deployment. Specify a policy by name or by ID, or use the **Get-CMAntimalwarePolicy** cmdlet to get a policy object. Specify a collection by name or by ID, or use the **Get-CMDeviceCollection** cmdlet to get a collection object. You can also specify a deployment name to deploy a policy.

Parameters

-AntimalwarePolicy<IResultObject>

Specifies an antimalware policy object in Configuration Manager. To obtain an antimalware policy object, use the **Get-CMAntiMalwarePolicy** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AntimalwarePolicyId<String>

Specifies the ID of an antimalware policy.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-AntimalwarePolicyName<String>

Specifies the name of an antimalware policy.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Collection<IResultObject>

Specifies a Configuration Manager device collection object. To obtain a device collection object, use the **Get-CMDeviceCollection** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CollectionId<String>

Specifies the ID of a device collection.

Aliases	none
Required?	true
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CollectionName<String>

Specifies the name of a device collection.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies the name of an antimalware policy deployment.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Deploy an antimalware policy to a collection by using an ID

This command starts the deployment for a policy named ExclusionsPolicy to a device collection that has the ID SMS00023.

```
PS C:\> Start-CMAntimalwarePolicyDeployment -AntimalwarePolicyName "ExclusionsPolicy" -CollectionId "SMS00023"
```

Example 2: Deploy an antimalware policy to a named collection

This command starts the deployment for a policy named ExclusionsPolicy to a device collection named SouthOffice.

```
PS C:\> Start-CMAntimalwarePolicyDeployment -AntimalwarePolicyName "ExclusionsPolicy" -CollectionName "SouthOffice"
```

Related topics

[Export-CMAntimalwarePolicy](#)

[Get-CMAntiMalwarePolicy](#)
[Merge-CMAntimalwarePolicy](#)
[New-CMAntimalwarePolicy](#)
[Remove-CMAntiMalwarePolicy](#)
[Set-CMAntiMalwarePolicy](#)
[Get-CMDeviceCollection](#)

Start-CMApplicationDeployment

Start-CMApplicationDeployment

Starts an application deployment in Configuration Manager.

Syntax

Parameter Set: SearchByNameMandatory

```
Start-CMApplicationDeployment -CollectionName <String> -Name <String[]> [-AppRequiresApproval <Boolean> ] [-AvaliableDate <DateTime> ] [-AvaliableTime <DateTime> ] [-Comment <String> ] [-DeadlineDate <DateTime> ] [-DeadlineTime <DateTime> ] [-DeployAction {Install | Uninstall} ] [-DeployPurpose {Available | Required} ] [-EnableMomAlert <Boolean> ] [-FailParameterValue <Int32> ] [-OverrideServiceWindow <Boolean> ] [-PersistOnWriteFilterDevice <Boolean> ] [-PostponeDate <DateTime> ] [-PostponeTime <DateTime> ] [-PreDeploy <Boolean> ] [-RaiseMomAlertsOnFailure <Boolean> ] [-RebootOutsideServiceWindow <Boolean> ] [-SendWakeUpPacket <Boolean> ] [-SuccessParameterValue <Int32> ] [-TimeBaseOn {LocalTime | UTC} ] [-UseMeteredNetwork <Boolean> ] [-UserNotification {DisplayAll | DisplaySoftwareCenterOnly | HideAll} ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Start-CMApplicationDeployment -CollectionName <String> -Id <String[]> [-AppRequiresApproval <Boolean> ] [-AvaliableDate <DateTime> ] [-AvaliableTime <DateTime> ] [-Comment <String> ] [-DeadlineDate <DateTime> ] [-DeadlineTime <DateTime> ] [-DeployAction {Install | Uninstall} ] [-DeployPurpose {Available | Required} ] [-EnableMomAlert <Boolean> ] [-FailParameterValue <Int32> ] [-OverrideServiceWindow <Boolean> ] [-PersistOnWriteFilterDevice <Boolean> ] [-PostponeDate <DateTime> ] [-PostponeTime <DateTime> ] [-PreDeploy <Boolean> ] [-RaiseMomAlertsOnFailure <Boolean> ] [-RebootOutsideServiceWindow <Boolean> ] [-SendWakeUpPacket <Boolean> ] [-SuccessParameterValue <Int32> ] [-TimeBaseOn {LocalTime | UTC} ] [-UseMeteredNetwork <Boolean> ] [-UserNotification {DisplayAll | DisplaySoftwareCenterOnly | HideAll} ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Start-CMApplicationDeployment -CollectionName <String> -InputObject <IResultObject> [-AppRequiresApproval <Boolean> ] [-AvaliableDate <DateTime> ] [-AvaliableTime <DateTime> ] [-Comment <String> ] [-DeadlineDate <DateTime> ] [-DeadlineTime <DateTime> ] [-DeployAction {Install | Uninstall} ] [-DeployPurpose {Available | Required} ] [-EnableMomAlert <Boolean> ] [-FailParameterValue <Int32> ] [-OverrideServiceWindow <Boolean> ] [-PersistOnWriteFilterDevice <Boolean> ] [-PostponeDate <DateTime> ] [-PostponeTime <DateTime> ] [-PreDeploy <Boolean> ] [-RaiseMomAlertsOnFailure <Boolean> ] [-RebootOutsideServiceWindow <Boolean> ] [-SendWakeUpPacket <Boolean> ] [-SuccessParameterValue <Int32> ] [-TimeBaseOn {LocalTime | UTC} ] [-UseMeteredNetwork <Boolean> ] [-UserNotification {DisplayAll | DisplaySoftwareCenterOnly | HideAll} ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Start-CMAApplicationDeployment** cmdlet starts an application deployment.

Parameters

-AppRequiresApproval<Boolean>

Specifies whether an application requires administrator approval prior to installation.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AvailableDate<DateTime>

Specifies a date when the application becomes available.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AvailableTime<DateTime>

Specifies a time when the application becomes available.

Aliases	none
Required?	false
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CollectionName<String>

Specifies a target collection to deploy this application.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Comment<String>

Specifies a comment.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeadlineDate<DateTime>

Specifies a day by which to install an application. Autoinstall performs the installation if the application is not installed.

Aliases	none
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeadlineTime<DateTime>

Specifies a time by which to install an application. Autoinstall performs the installation if the application is not installed.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeployAction<DeployActionType>

Specifies an action for a deployment. Valid values are:

- Install. Install the application.
- Uninstall. Uninstall the application.

The acceptable values for this parameter are:

Install	
Uninstall	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-DeployPurpose<DeployPurposeType>

Specifies a deployment purpose. Valid values are:

-- Available. If the target collection is a device collection, the application is available in the software center. If the target collection is a user collection, the application is available in the catalog web site.

-- Required. Installation occurs when the deadline passes.

The acceptable values for this parameter are:

Available	
Required	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnableMomAlert<Boolean>

Specifies whether to enable Operations Manager maintenance mode.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-FailParameterValue<Int32>

Specifies a value that generates a deployment alert when exceeded.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Id<String[]>

Specifies an array of IDs.

Aliases	CId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies an application deployment object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String[]>

Specifies an array of names.

Aliases	LocalizedDisplayName
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OverrideServiceWindow<Boolean>

Specifies whether an application installation occurs outside of a maintenance window.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PersistOnWriteFilterDevice<Boolean>

Specifies whether to commit changes on a Windows Embedded device at deadline or during a maintenance window. Otherwise, changes are written on the overlay and committed later.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PostponeDate<DateTime>

Specifies a date after which to create an alert.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PostponeTime<DateTime>

Specifies a time after which to create an alert.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PreDeploy<Boolean>

Specifies whether to copy software to a device prior to installation. To use this parameter, set the *DeployPurpose* parameter to Required.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RaiseMomAlertsOnFailure<Boolean>

Specifies whether to raise Operations Manager alerts when a failure occurs.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RebootOutsideServiceWindow<Boolean>

Indicates whether a computer restarts outside a service window. A service window is a specified period of time used for computer maintenance and updates. If this value is \$True, any required restart takes place without regard to service windows. If this value is \$False, the computer does not restart outside a service window.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SendWakeUpPacket<Boolean>

Indicates whether to send a wake up packet to computers before the deployment begins. If this value is \$True, Configuration Manager wakes a computer from sleep. If this value is \$False, it does not wake computers from sleep. For computers to wake, you must first configure Wake On LAN.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SuccessParameterValue<Int32>

Specifies a value that the threshold must exceed before an alert is created.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-TimeBaseOn<TimeType>

Specifies which time zone to use. Valid values are:

-- LocalTime. Use local time.

-- UTC. Use Coordinated Universal Time (UTC), also known as Greenwich Mean Time.

The acceptable values for this parameter are:

LocalTime	
UTC	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UseMeteredNetwork<Boolean>

Specifies whether clients can download content over metered Internet connections after the installation deadline. Clients may incur additional costs.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UserNotification<UserNotificationType>

Specifies user notification types. Valid values are:

- DisplayAll. Display in Software Center and show all notifications.
- DisplaySoftwareCenterOnly. Display in Software Center and only show notifications for computer restarts.
- HideAll. Do not display in Software Center and do not show notifications.

The acceptable values for this parameter are:

DisplayAll	
DisplaySoftwareCenterOnly	
HideAll	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Start application deployment

This command starts an application deployment named 7zip.

```
PS C:\> Start-CMApplicationDeployment -CollectionName "All Users" -Name "7zip" -
AvaliableDate 2012/10/1 -AvaliableTime 12:45 -Comment "test" -DeadlineDate 2013/10/23 -
DeadlineTime 21:12 -DeployAction Uninstall -EnableMomAlert $True -FailParameterValue 40 -
OverrideServiceWindow $True -PersistOnWriteFilterDevice $False -PostponeDate 2014/2/8 -
PostponeTime 11:11 -PreDeploy $True -RaiseMomAlertsOnFailure $True -
RebootOutsideServiceWindow $True -SendWakeUpPacket $True -SuccessParameterValue 30 -
UseMeteredNetwork $True -UserNotification DisplaySoftwareCenterOnly
```

Related topics

[Set-CMApplicationDeployment](#)

Start-CMApplicationDeploymentSimulation

Start-CMApplicationDeploymentSimulation

Starts an application deployment simulation in Configuration Manager.

Syntax

Parameter Set: SearchByNameMandatory

```
Start-CMApplicationDeploymentSimulation -CollectionName <String> -Name <String[]> [-DeployAction {Install | Uninstall} ] [-PreDeploy <Boolean> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Start-CMApplicationDeploymentSimulation -CollectionName <String> -Id <String[]> [-DeployAction {Install | Uninstall} ] [-PreDeploy <Boolean> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Start-CMApplicationDeploymentSimulation -CollectionName <String> -InputObject <IResultObject> [-DeployAction {Install | Uninstall} ] [-PreDeploy <Boolean> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Start-CMApplicationDeploymentSimulation** cmdlet starts an application deployment. Use simulated deployment to test an application deployment without installing an application.

Parameters

-CollectionName<String>

Specifies a name for the target collection.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-DeployAction<DeployActionType>

Specifies the action to take. Valid values are:

-- Install. Install the application.

-- Uninstall. Uninstall the application.

The acceptable values for this parameter are:

Install	
Uninstall	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Id<String[]>

Specifies an array of IDs.

Aliases	CId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies an application deployment object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String[]>

Specifies an array of names.

Aliases	LocalizedDisplayName
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PreDeploy<Boolean>

Specifies whether to copy software to a device prior to installation.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Start an application deployment simulation

This command starts a deployment simulation of the installation of the application named WIN8_UPDATE2 for the target collection named All Mobile Devices.

```
PS C:\> Start-CMApplicationDeploymentSimulation -CollectionName "All Mobile Devices" -Name "WIN8_UPDATE2" -DeployAction Install
```

Related topics

[Start-CMApplicationDeployment](#)

[Set-CMApplicationDeployment](#)

Start-CMBaselineDeployment

Start-CMBaselineDeployment

Starts deployment of a Configuration Manager baseline configuration to a collection of computers.

Syntax

Parameter Set: SearchByNameMandatory

```
Start-CMBaselineDeployment -CollectionName <String> -Name <String[]> [-EnableEnforcement <Boolean> ] [-GenerateAlert <Boolean> ] [-MonitoredByScm <Boolean> ] [-OverrideServiceWindow <Boolean> ] [-ParameterValue <Int32> ] [-PostponeDate <DateTime> ] [-PostponeTime <DateTime> ] [-Schedule <IResultObject> ] [-Confirm] [-WhatIf] [<CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Start-CMBaselineDeployment -CollectionName <String> -Id <String[]> [-EnableEnforcement <Boolean> ] [-GenerateAlert <Boolean> ] [-MonitoredByScm <Boolean> ] [-OverrideServiceWindow <Boolean> ] [-ParameterValue <Int32> ] [-PostponeDate <DateTime> ] [-PostponeTime <DateTime> ] [-Schedule <IResultObject> ] [-Confirm] [-WhatIf] [<CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Start-CMBaselineDeployment -CollectionName <String> -InputObject <IResultObject> [-EnableEnforcement <Boolean> ] [-GenerateAlert <Boolean> ] [-MonitoredByScm <Boolean> ] [-OverrideServiceWindow <Boolean> ] [-ParameterValue <Int32> ] [-PostponeDate <DateTime> ] [-PostponeTime <DateTime> ] [-Schedule <IResultObject> ] [-Confirm] [-WhatIf] [<CommonParameters>]
```

Detailed Description

The **Start-CMBaselineDeployment** cmdlet starts the deployment of a Microsoft System Center 2012 Configuration Manager baseline configuration to a collection of computers.

A baseline defines the configuration of a product or system established at a specific time. Baselines contain a defined set of required configurations and associated rules. System Center 2012 Configuration Manager assigns baselines to computer in collections, together with a compliance evaluation schedule.

Parameters

-CollectionName<String>

Specifies a name of a collection. The deployment applies to this collection.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnableEnforcement<Boolean>

Specifies whether to enable enforcement for the deployment. During enforcement, a client reports compliance information about a deployment.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-GenerateAlert<Boolean>

Specifies whether Configuration Manager generates alerts during the deployment.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-Id<String[]>

Specifies an array of IDs of baseline deployments.

Aliases	CIId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a baseline deployment object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-MonitoredByScom<Boolean>

Specifies whether to apply System Center 2012 – Operations Manager monitoring criteria during the deployment.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String[]>

Specifies an array of names for baseline deployments.

Aliases	LocalizedDisplayName
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OverrideServiceWindow<Boolean>

Specifies whether to override service windows while deploying policies. Service windows are periods of time allocated for maintenance.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ParameterValue<Int32>

Specifies an integer value. This is the parameter value.

Aliases	none
Required?	false
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PostponeDate<DateTime>

Specifies a date, as a **DateTime** object. To obtain a **DateTime** object, use the **Get-Date** cmdlet. For more information, type `Get-Help Get-Date`. This is the date for the deployment if postponed.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PostponeTime<DateTime>

Specifies a time, as a **DateTime** object. To obtain a **DateTime** object, use the **Get-Date** cmdlet. This is the time for the deployment if postponed.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Schedule<IResultObject>

Specifies a schedule object. This is the schedule for evaluating the baseline.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Start baseline deployment

This command starts a baseline deployment named Baseline22 for the collection All Users. The command enables enforcement, generates alerts, monitors the deployment using Operations Manager, and overrides defined service windows. The command specifies a parameter value of 30.

```
PS C:\> Start-CMBaselineDeployment -CollectionName "All Users" -Name "Baseline22" -  
EnableEnforcement $True -GenerateAlert $True -MonitoredByScom $True -OverrideServiceWindow  
$True -ParameterValue 30
```

Related topics

[Set-CMBaselineDeployment](#)

Start-CMClientSettingDeployment

Start-CMClientSettingDeployment

Deploys client settings to devices in a collection.

Syntax

Parameter Set: SearchByClientSettingId_CollectionId

```
Start-CMClientSettingDeployment -ClientSettingId <String> -CollectionId <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByClientSettingId_CollectionName

```
Start-CMClientSettingDeployment -ClientSettingId <String> -CollectionName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByClientSettingId_CollectionValue

```
Start-CMClientSettingDeployment -ClientSettingId <String> -Collection <IResultObject> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByClientSettingName_CollectionId

```
Start-CMClientSettingDeployment -ClientSettingName <String> -CollectionId <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByClientSettingName_CollectionName

```
Start-CMClientSettingDeployment -ClientSettingName <String> -CollectionName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByClientSettingName_CollectionValue

```
Start-CMClientSettingDeployment -ClientSettingName <String> -Collection <IResultObject> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByClientSettingValue_CollectionId

```
Start-CMClientSettingDeployment -ClientSetting <IResultObject> -CollectionId <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByClientSettingValue_CollectionName

```
Start-CMClientSettingDeployment -ClientSetting <IResultObject> -CollectionName <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByClientSettingValue_CollectionValue

```
Start-CMClientSettingDeployment -ClientSetting <IResultObject> -Collection <IResultObject> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Start-CMClientSettingDeployment** cmdlet deploys client settings to devices in a Microsoft System Center 2012 Configuration Manager collection. Specify the client setting object by using its name or ID, or you can use the **Get-CMClientSetting** cmdlet to get a client setting object. Specify the collection to apply the settings to by using its name or ID, or you can use the **Get-CMDeviceCollection** cmdlet to get a device collection.

For more information about client settings, see [About Client Settings in Configuration Manager](http://go.microsoft.com/fwlink/?LinkId=266226) (<http://go.microsoft.com/fwlink/?LinkId=266226>) on TechNet.

Parameters

-ClientSetting<IResultObject>

Specifies a client setting object. To obtain a client setting object, use the **Get-CMClientSetting** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ClientSettingId<String>

Specifies the ID of a client setting object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ClientSettingName<String>

Specifies the name of a client setting object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Collection<IResultObject>

Specifies a Configuration Manager collection object. To obtain a collection object, use the **Get-CMDeviceCollection** cmdlet. Configuration Manager applies the client settings to the members of this collection.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CollectionId<String>

Specifies the ID of a Configuration Manager collection. Configuration Manager applies the client settings to the members of this collection.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CollectionName<String>

Specifies the name of a Configuration Manager collection. Configuration Manager applies the client settings to the members of this collection.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Deploy a client setting object by using its ID to a named collection

This command starts deployment of the client setting object that has the ID CSI1023 for the collection named General Computer Collection.

```
PS C:\> Start-CMClientSettingDeployment -ClientSettingId "CSI1023" -CollectionName "General Computer Collection"
```

Example 2: Deploy a client setting object by using a variable

The first command gets a client setting object that has the ID CSI1023, and saves it in the \$CSID variable.

The second command starts deployment of the client setting object in the \$CSID variable to the collection named General Computer Collection.

```
PS C:\> $CSID = Get-CMClientSetting -Id "CSI1023"  
PS C:\> Start-CMClientSettingDeployment -ClientSetting $CSID -CollectionName "General Computer Collection"
```

Related topics

[Get-CMClientSetting](#)

[Get-CMDeviceCollection](#)

Start-CMCloudDistributionPoint

Start-CMCloudDistributionPoint

Starts the cloud distribution point service.

Syntax

Parameter Set: SearchByIdMandatory

```
Start-CMCloudDistributionPoint -Id <String[]> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByNameMandatory

```
Start-CMCloudDistributionPoint -Name <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Start-CMCloudDistributionPoint -InputObject <IResultObject> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Start-CMCloudDistributionPoint** cmdlet starts the cloud distribution point service.

You can use the **Stop-CMCloudDistributionPoint** cmdlet to suspend distribution of content.

Parameters

-Id<String[]>

Specifies an array of identifiers for cloud distribution points. You can use a comma separated list.

Aliases	AzureServiceId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a cloud distribution point object. To obtain a cloud distribution point object, you can use the **Get-CMCloudDistributionPoint** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies the name of a cloud distribution point.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Start the cloud distribution point service using an ID

This command starts the cloud distribution point service for the cloud distribution point that has the specified identifier.

```
PS C:\> Start-CMCloudDistributionPoint -Id "16777242"
```

Example 2: Start the cloud distribution point service using a name

This command starts the cloud distribution point service for the cloud distribution point named West01.

```
PS C:\> Start-CMCloudDistributionPoint -Name "West01"
```

Example 3: Start the cloud distribution point service using an object

The first command uses the **Get-CMCloudDistributionPoint** cmdlet to get the distribution point with the specified identifier, and then saves it in the \$DistPnt variable.

The second command starts the cloud distribution point service for the distribution point stored in \$DistPnt.

```
PS C:\> $DistPnt = Get-CMCloudDistributionPoint -Id "16777242"  
PS C:\> Start-CMCloudDistributionPoint -InputObject $DistPnt
```

Related topics

[Get-CMCloudDistributionPoint](#)

[New-CMCloudDistributionPoint](#)

[Remove-CMCloudDistributionPoint](#)

[Set-CMCloudDistributionPoint](#)

[Stop-CMCloudDistributionPoint](#)

Start-CMConfigurationPolicyDeployment

Start-CMConfigurationPolicyDeployment

Deploys policies for a Configuration Manager collection.

Syntax

Parameter Set: DeployFWPolicyByIdMandatory

```
Start-CMConfigurationPolicyDeployment -CollectionName <String> -FirewallPolicyId <String> [-Schedule <IResultObject> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: DeployFWPolicyByNameMandatory

```
Start-CMConfigurationPolicyDeployment -CollectionName <String> -FirewallPolicyName <String> [-Schedule <IResultObject> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: DeployFWPolicyByValueMandatory

```
Start-CMConfigurationPolicyDeployment -CollectionName <String> -FirewallPolicy <IResultObject> [-Schedule <IResultObject> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: DeployRCPolicyByIdMandatory

```
Start-CMConfigurationPolicyDeployment -CollectionName <String> -RemoteConnectionProfileId <String> [-EnableEnforcement <Boolean> ] [-GenerateAlert <Boolean> ] [-MonitoredByScom <Boolean> ] [-OverrideServiceWindow <Boolean> ] [-ParameterValue <Int32> ] [-PostponeDate <DateTime> ] [-PostponeTime <DateTime> ] [-Schedule <IResultObject> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: DeployRCPolicyByNameMandatory

```
Start-CMConfigurationPolicyDeployment -CollectionName <String> -RemoteConnectionProfileName <String> [-EnableEnforcement <Boolean> ] [-GenerateAlert <Boolean> ] [-MonitoredByScom <Boolean> ] [-OverrideServiceWindow <Boolean> ] [-ParameterValue <Int32> ] [-PostponeDate <DateTime> ] [-PostponeTime <DateTime> ] [-Schedule <IResultObject> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: DeployRCPolicyByValueMandatory

```
Start-CMConfigurationPolicyDeployment -CollectionName <String> -RemoteConnectionProfile <IResultObject> [-EnableEnforcement <Boolean> ] [-GenerateAlert <Boolean> ] [-MonitoredByScom <Boolean> ] [-OverrideServiceWindow <Boolean> ] [-ParameterValue <Int32> ] [-PostponeDate <DateTime> ] [-PostponeTime <DateTime> ] [-Schedule <IResultObject> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: DeployUSMPolicyByIdMandatory

```
Start-CMConfigurationPolicyDeployment -CollectionName <String> -UserDataAndProfileId <String> [-EnableEnforcement <Boolean> ] [-GenerateAlert <Boolean> ] [-MonitoredByScom <Boolean> ] [-OverrideServiceWindow <Boolean> ] [-ParameterValue <Int32> ] [-PostponeDate <DateTime> ] [-PostponeTime <DateTime> ] [-Schedule <IResultObject> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: DeployUSMPolicyByNameMandatory

```
Start-CMConfigurationPolicyDeployment -CollectionName <String> -UserDataAndProfileName <String> [-EnableEnforcement <Boolean> ] [-GenerateAlert <Boolean> ] [-MonitoredByScom <Boolean> ] [-OverrideServiceWindow <Boolean> ] [-ParameterValue <Int32> ] [-PostponeDate <DateTime> ] [-PostponeTime <DateTime> ] [-Schedule <IResultObject> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: DeployUSMPolicyByValueMandatory

```
Start-CMConfigurationPolicyDeployment -CollectionName <String> -UserDataAndProfile <IResultObject> [-EnableEnforcement <Boolean> ] [-GenerateAlert <Boolean> ] [-MonitoredByScom <Boolean> ] [-OverrideServiceWindow <Boolean> ] [-ParameterValue <Int32> ] [-PostponeDate <DateTime> ] [-PostponeTime <DateTime> ] [-Schedule <IResultObject> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Start-CMConfigurationPolicyDeployment** cmdlet deploys specified policies for a Microsoft System Center 2012 Configuration Manager collection. You can deploy firewall policies or user session management policies.

You can specify a firewall policy by name or by ID or use another cmdlet to get firewall policy object.

You can specify System Center 2012 – Operations Manager monitoring criteria.

Parameters

-CollectionName<String>

Specifies the name of a collection. The deployment applies to this Configuration Manager collection.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnableEnforcement<Boolean>

Specifies whether to enable enforcement for the deployment. During enforcement, a client reports compliance information about a deployment.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-FirewallPolicy<IResultObject>

Specifies a firewall policy object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-FirewallPolicyId<String>

Specifies an ID for a firewall policy.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-FirewallPolicyName<String>

Specifies a name for a firewall policy.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-GenerateAlert<Boolean>

Specifies whether Configuration Manager generates alerts during the deployment.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-MonitoredByScom<Boolean>

Specifies whether Operations Manager monitoring criteria applies during the deployment.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OverrideServiceWindow<Boolean>

Specifies whether to override the service window while deploying policies.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ParameterValue<Int32>

Specifies an integer value. This is the parameter value.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PostponeDate<DateTime>

Specifies a date, as a **DateTime** object. To obtain a **DateTime** object, use the **Get-Date** cmdlet. For more information, type `Get-Help Get-Date`. This is the date for the deployment if postponed.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PostponeTime<DateTime>

Specifies a time, as a **DateTime** object. To obtain a **DateTime** object, use the **Get-Date** cmdlet. This is the time for the deployment if postponed.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RemoteConnectionProfile<IResultObject>

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RemoteConnectionProfileId<String>

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RemoteConnectionProfileName<String>

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Schedule<IResultObject>

Specifies a schedule object. This is the schedule for evaluating the policy.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UserDataAndProfile<IResultObject>

Specifies a user data and profile object. To obtain a user data and profile object, use the **Get-CMUserDataAndProfileConfigurationItem** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UserDataAndProfileId<String>

Specifies an ID for a user data and profile object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UserDataAndProfileName<String>

Specifies a name for a user data and profile object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Start deployment of a firewall policy

This command starts the configuration policy deployment for a collection named Desktop systems. The command specifies a firewall policy named General firewall policy.

```
PS C:\> Start-CMConfigurationPolicyDeployment -CollectionName "Desktop systems" -  
FirewallPolicyName "General firewall policy"
```

Related topics

[Set-CMConfigurationPolicyDeployment](#)

[Get-CMUserDataAndProfileConfigurationItem](#)

Start-CMContentDistribution

Start-CMContentDistribution

Copies content to distribution points.

Syntax

Parameter Set: SearchByIdMandatory_Application

```
Start-CMContentDistribution -ApplicationId <String[]> [-CollectionName <String> ] [-DisableDetectAssociatedContentDependencies] [-DistributionPointGroupName <String> ] [-DistributionPointName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory_BootImage

```
Start-CMContentDistribution -BootImageId <String[]> [-CollectionName <String> ] [-DistributionPointGroupName <String> ] [-DistributionPointName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory_DeploymentPackage

```
Start-CMContentDistribution -DeploymentPackageId <String[]> [-CollectionName <String> ] [-DistributionPointGroupName <String> ] [-DistributionPointName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory_DriverPackage

```
Start-CMContentDistribution -DriverPackageId <String[]> [-CollectionName <String> ] [-DistributionPointGroupName <String> ] [-DistributionPointName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory_OperatingSystemImage

```
Start-CMContentDistribution -OperatingSystemImageId <String[]> [-CollectionName <String> ] [-DistributionPointGroupName <String> ] [-DistributionPointName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory_OperatingSystemInstaller

```
Start-CMContentDistribution -OperatingSystemInstallerId <String[]> [-CollectionName <String> ] [-DistributionPointGroupName <String> ] [-DistributionPointName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory_Package

```
Start-CMContentDistribution -PackageId <String[]> [-CollectionName <String> ] [-DistributionPointGroupName <String> ] [-DistributionPointName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory_TaskSequence

```
Start-CMContentDistribution -TaskSequenceId <String[]> [-CollectionName <String> ] [-DistributionPointGroupName <String> ] [-DistributionPointName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByNameMandatory_Application

Start-CMContentDistribution -ApplicationName <String[]> [-CollectionName <String>] [-DisableDetectAssociatedContentDependencies] [-DistributionPointGroupName <String>] [-DistributionPointName <String>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SearchByNameMandatory_BootImage

Start-CMContentDistribution -BootImageName <String[]> [-CollectionName <String>] [-DistributionPointGroupName <String>] [-DistributionPointName <String>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SearchByNameMandatory_DeploymentPackage

Start-CMContentDistribution -DeploymentPackageName <String[]> [-CollectionName <String>] [-DistributionPointGroupName <String>] [-DistributionPointName <String>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SearchByNameMandatory_DriverPackage

Start-CMContentDistribution -DriverPackageName <String[]> [-CollectionName <String>] [-DistributionPointGroupName <String>] [-DistributionPointName <String>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SearchByNameMandatory_OperatingSystemImage

Start-CMContentDistribution -OperatingSystemImageName <String[]> [-CollectionName <String>] [-DistributionPointGroupName <String>] [-DistributionPointName <String>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SearchByNameMandatory_OperatingSystemInstaller

Start-CMContentDistribution -OperatingSystemInstallerName <String[]> [-CollectionName <String>] [-DistributionPointGroupName <String>] [-DistributionPointName <String>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SearchByNameMandatory_Package

Start-CMContentDistribution -PackageName <String[]> [-CollectionName <String>] [-DistributionPointGroupName <String>] [-DistributionPointName <String>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SearchByNameMandatory_TaskSequence

Start-CMContentDistribution -TaskSequenceName <String[]> [-CollectionName <String>] [-DistributionPointGroupName <String>] [-DistributionPointName <String>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SearchByValueMandatory_Application

Start-CMContentDistribution -Application <IResultObject> [-CollectionName <String>] [-DisableDetectAssociatedContentDependencies] [-DistributionPointGroupName <String>] [-DistributionPointName <String>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SearchByValueMandatory_BootImage

Start-CMContentDistribution -BootImage <IResultObject> [-CollectionName <String>] [-DistributionPointGroupName <String>] [-DistributionPointName <String>] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SearchByValueMandatory_DeploymentPackage

Start-CMContentDistribution -DeploymentPackage <IResultObject> [-CollectionName <String>]

```
[-DistributionPointGroupName <String> ] [-DistributionPointName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory_DriverPackage

```
Start-CMContentDistribution -DriverPackage <IResultObject> [-CollectionName <String> ] [-DistributionPointGroupName <String> ] [-DistributionPointName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory_OperatingSystemImage

```
Start-CMContentDistribution -OperatingSystemImage <IResultObject> [-CollectionName <String> ] [-DistributionPointGroupName <String> ] [-DistributionPointName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory_OperatingSystemInstaller

```
Start-CMContentDistribution -OperatingSystemInstaller <IResultObject> [-CollectionName <String> ] [-DistributionPointGroupName <String> ] [-DistributionPointName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory_Package

```
Start-CMContentDistribution -Package <IResultObject> [-CollectionName <String> ] [-DistributionPointGroupName <String> ] [-DistributionPointName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory_TaskSequence

```
Start-CMContentDistribution -TaskSequence <IResultObject> [-CollectionName <String> ] [-DistributionPointGroupName <String> ] [-DistributionPointName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Start-CMContentDistribution** cmdlet copies content from the content library on a site server to the content library on the distribution points.

You can use this cmdlet to distribute several types of content, including application deployment types, packages, deployment packages, driver packages, operating system images, operating system installers, boot images, and task sequences. You can distribute the content to distribution points, distribution point groups, or collections associated with distribution point groups.

Parameters

-Application<IResultObject>

Specifies a Configuration Manager application object. To obtain a **CMApplication** object, use the **Get-CMApplication** cmdlet.

Aliases	none
---------	------

Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ApplicationId<String[]>

Specifies an array of application IDs.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ApplicationName<String[]>

Specifies an array of application names.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-BootImage<IResultObject>

Specifies a boot image object. To obtain a **CMBootImage** object, use the **Get-CMBootImage** cmdlet.

Aliases	none
---------	------

Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-BootImageId<String[]>

Specifies an array of IDs of boot images.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-BootImageName<String[]>

Specifies an array of names of boot images.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CollectionName<String>

Specifies the name of a Configuration Manager collection.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeploymentPackage<IResultObject>

Specifies a deployment package object. To obtain a **CMDeploymentPackage** object, use the **Get-
CMDeploymentPackage** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeploymentPackageId<String[]>

Specifies an array of IDs of deployment packages.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeploymentPackageName<String[]>

Specifies an array of names of deployment packages.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DisableDetectAssociatedContentDependencies

Indicates that Configuration Manager automatically detects associated content dependencies and adds the associated content to the distribution for applications.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DistributionPointGroupName<String>

Specifies the name of a distribution point group.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DistributionPointName<String>

Specifies the name of a distribution point that is associated with the deployment package.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DriverPackage<IResultObject>

Specifies a driver package object. To obtain a **CMDriverPackage** object, use the **Get-CMDriverPackage** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DriverPackageld<String[]>

Specifies an array of IDs of driver packages.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DriverPackageName<String[]>

Specifies an array of names of driver packages.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OperatingSystemImage<IResultObject>

Specifies an operating system image object. To obtain a **CMOperatingSystemImage** object, use the **Get-CMOperatingSystemImage** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OperatingSystemImageId<String[]>

Specifies an array of IDs of operating system images.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OperatingSystemImageName<String[]>

Specifies an array of names of operating system images.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OperatingSystemInstaller<IResultObject>

Specifies an operating system installer object. To obtain a **CMOperatingSystemInstaller** object, use the **Get-CMOperatingSystemInstaller** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OperatingSystemInstallerId<String[]>

Specifies an array of IDs of operating system installers.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OperatingSystemInstallerName<String[]>

Specifies an array of names of operating system installers.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Package<IResultObject>

Specifies a package object. To obtain a **CMPackage** object, use the **Get-CMPackage** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Packageld<String[]>

Specifies an array of IDs of packages.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PackageName<String[]>

Specifies an array of names of packages.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-TaskSequence<IResultObject>

Specifies a task sequence object. To obtain a **CMTaskSequence** object, use the **Get-CMTaskSequence** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-TaskSequenceId<String[]>

Specifies an array of IDs of task sequences.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-TaskSequenceName<String[]>

Specifies an array of names of task sequences.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Distribute a boot image

This command distributes the boot image that has the ID CM200004. The command distributes the boot image to the collection named All Systems, the distribution point named CMDIV-TSQA04.CORP.CONTOSO.COM, and the distribution point group named DistPtGroup02.

```
PS C:\> Start-CMContentDistribution -BootImageId "CM200004" -CollectionName "All Systems" -
DistributionPointName "CMDIV-TSQA04.CORP.CONTOSO.COM" -DistributionPointGroupName
"DistPtGroup02"
```

Example 2: Distribute a task sequence

This command distributes the task sequence that has the ID CM200007 to the collection named All Systems and the distribution point named CMDIV-TSQA04.CORP.CONTOSO.COM.

```
PS C:\> Start-CMContentDistribution -TaskSequenceId "CM200007" -CollectionName "All Systems" -
-DistributionPointName "CMDIV-TSQA04.CORP.CONTOSO.COM"
```

Example 3: Distribute an application

This command distributes the application named Dict.app. The command distributes the application to the collection named All Systems, the distribution point named CMDIV-TSQA04.CORP.CONTOSO.COM, and the distribution point group named DistPtGroup02.

```
PS C:\> Start-CMContentDistribution -ApplicationName "Dict.app" -CollectionName "All
Systems" -DistributionPointName "CMDIV-TSQA04.CORP.CONTOSO.COM" -DistributionPointGroupName
"DistPtGroup02"
```

Example 4: Distribute a package

This command distributes the package that has the ID CM200001. The command distributes the package to the collection named All Systems, the distribution point named CMDIV-TSQA04.CORP.CONTOSO.COM, and the distribution point group named DistPtGroup02.

```
PS C:\> Start-CMContentDistribution -PackageId "CM200001" -CollectionName "All Systems" -
DistributionPointName "CMDIV-TSQA04.CORP.CONTOSO.COM" -DistributionPointGroupName
"DistPtGroup02"
```

Example 5: Distribute a deployment package

This command distributes the deployment package named DivDeployPkg01. The command distributes the deployment package to the collection named All Systems, the distribution point named CMDIV-TSQA04.CORP.CONTOSO.COM, and the distribution point group named DistPtGroup02.

```
PS C:\> Start-CMContentDistribution -DeploymentPackageName "DivDeployPkg01" -CollectionName "All Systems" -DistributionPointName "CMDIV-TSQA04.CORP.CONTOSO.COM" -DistributionPointGroupName "DistPtGroup02"
```

Example 6: Distribute a driver package

This command distributes the driver package named DrvPkg02. The command distributes the driver package to the collection named All Systems, the distribution point named CMDIV-TSQA04.CORP.CONTOSO.COM, and the distribution point group named DistPtGroup02.

```
PS C:\> Start-CMContentDistribution -DriverPackageName "DrvPkg02" -CollectionName "All Systems" -DistributionPointName "CMDIV-TSQA04.CORP.CONTOSO.COM" -DistributionPointGroupName "DistPtGroup02"
```

Example 7: Distribute an operating system image

This command distributes the operating system image that has the ID CM200013. The command distributes the operating system image to the collection named All Systems, the distribution point named CMDIV-TSQA04.CORP.CONTOSO.COM, and the distribution point group named DistPtGroup02.

```
PS C:\> Start-CMContentDistribution -OperatingSystemImageId "CM200013" -CollectionName "All Systems" -DistributionPointName "CMDIV-TSQA04.CORP.CONTOSO.COM" -DistributionPointGroupName "DistPtGroup02"
```

Example 8: Distribute an operating system installer

This command distributes the operating system installer that has the ID CM200017. The command distributes the operating system installer to the collection named All Systems, the distribution point named CMDIV-TSQA04.CORP.CONTOSO.COM, and the distribution point group named DistPtGroup02.

```
PS C:\> Start-CMContentDistribution -OperatingSystemInstallerId "CM200017" -CollectionName "All Systems" -DistributionPointName "CMDIV-TSQA04.CORP.CONTOSO.COM" -DistributionPointGroupName "DistPtGroup02"
```

Related topics

[Remove-CMContentDistribution](#)

[Update-CMDistributionPoint](#)

[Get-CMOperatingSystemImage](#)

[Get-CMOperatingSystemInstaller](#)

[Get-CMApplication](#)

[Get-CMBootImage](#)

[Get-CMDeploymentPackage](#)

[Get-CMDriverPackage](#)

[Get-CMPackage](#)

[Get-CMTaskSequence](#)

Start-CMDistributionPointUpgrade

Start-CMDistributionPointUpgrade

Upgrades a shared distribution point.

Syntax

Parameter Set: StartDistributionPointUpgradeByImportCertificate

```
Start-CMDistributionPointUpgrade -AllowPreStaging <Boolean> -CertificatePath <String> -
SharedDistributionPoint <IResultObject> -SiteCode <String> [-AllowFallbackForContent
<Boolean> ] [-AllowRespondIncomingPxeRequest <Boolean> ] [-BoundaryGroup <IResultObject[]> ]
[-CertificatePassword <SecureString> ] [-ClientConnectionType {Internet |
InternetAndIntranet | Intranet} ] [-ComputersUsePxePassword <SecureString> ] [-
ContentValidationPriority {High | Highest | Low | Lowest | Medium} ] [-EnableAnonymous
<Boolean> ] [-EnablePxeSupport <Boolean> ] [-EnableUnknownComputerSupport <Boolean> ] [-
ForceWhenDuplicateCertificate <Boolean> ] [-InitiateConnection <Boolean> ] [-
InstallationAccount <IResultObject> ] [-InstallInternetServer <Boolean> ] [-
MacAddressForRespondingPxeRequest <String[]> ] [-MinFreeSpaceMB <Int32> ] [-
PathForSavingMigratedPackage <String> ] [-PrimaryContentLibraryLocation {A | Automatic | B |
C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y |
Z} ] [-PrimaryPackageShareLocation {A | Automatic | B | C | D | E | F | G | H | I | J | K |
L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z} ] [-PublicFqdn <String> ] [-
PxeServerResponseDelaySeconds <Int32> ] [-SecondaryContentLibraryLocation {A | Automatic | B
| C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y
| Z} ] [-SecondaryPackageShareLocation {A | Automatic | B | C | D | E | F | G | H | I | J |
K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z} ] [-UserDeviceAffinity
{AllowWithAutomaticApproval | AllowWithManualApproval | DoNotUse} ] [-
ValidateContentSchedule <IResultObject> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: StartDistributionPointUpgradeBySelfSignedCertificate

```
Start-CMDistributionPointUpgrade -AllowPreStaging <Boolean> -CertificateExpirationTimeUtc
<DateTime> -SharedDistributionPoint <IResultObject> -SiteCode <String> [-
AllowFallbackForContent <Boolean> ] [-AllowRespondIncomingPxeRequest <Boolean> ] [-
BoundaryGroup <IResultObject[]> ] [-ClientConnectionType {Internet | InternetAndIntranet |
Intranet} ] [-ComputersUsePxePassword <SecureString> ] [-ContentValidationPriority {High |
Highest | Low | Lowest | Medium} ] [-EnableAnonymous <Boolean> ] [-EnablePxeSupport
<Boolean> ] [-EnableUnknownComputerSupport <Boolean> ] [-InitiateConnection <Boolean> ] [-
InstallationAccount <IResultObject> ] [-InstallInternetServer <Boolean> ] [-
MacAddressForRespondingPxeRequest <String[]> ] [-MinFreeSpaceMB <Int32> ] [-
PathForSavingMigratedPackage <String> ] [-PrimaryContentLibraryLocation {A | Automatic | B |
C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y |
Z} ] [-PrimaryPackageShareLocation {A | Automatic | B | C | D | E | F | G | H | I | J | K |
L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z} ] [-PublicFqdn <String> ] [-
PxeServerResponseDelaySeconds <Int32> ] [-SecondaryContentLibraryLocation {A | Automatic | B
| C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y
| Z} ] [-SecondaryPackageShareLocation {A | Automatic | B | C | D | E | F | G | H | I | J |
K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z} ] [-UserDeviceAffinity
```

```
{AllowWithAutomaticApproval | AllowWithManualApproval | DoNotUse} ] [-  
ValidateContentSchedule <IResultObject> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Start-CMDistributionPointUpgrade** cmdlet upgrades a shared distribution point to a Microsoft System Center 2012 Configuration Manager distribution point.

When you migrate from a Microsoft System Center Configuration Manager 2007 source hierarchy, you can upgrade a shared distribution point to make it a System Center 2012 Configuration Manager distribution point. You can upgrade distribution points at both primary sites and secondary sites. The upgrade process removes the distribution point from the Configuration Manager 2007 hierarchy and makes it a site system server in the System Center 2012 Configuration Manager hierarchy. This process also copies the existing content that is on the distributing point to a new location on the distribution point computer. The upgrade process then modifies the copy of the content to create the System Center 2012 Configuration Manager single instance store for use with System Center 2012 Configuration Manager content deployment. Therefore, when you upgrade a distribution point, you do not have to redistribute migrated content that was hosted on the Configuration Manager 2007 distribution point.

Parameters

-AllowFallbackForContent<Boolean>

Indicates whether clients can use a fallback source location for content.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AllowPreStaging<Boolean>

Indicates whether the distribution point can pre-stage contents.

Aliases	none
---------	------

Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AllowRespondIncomingPxeRequest<Boolean>

Indicates whether the distribution point can respond to pre-boot execution environment (PXE) requests.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-BoundaryGroup<IResultObject[]>

Specifies an array of boundary group objects. To obtain a **CMBoundaryGroup** object, use the **Get-CMBoundaryGroup** cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CertificateExpirationTimeUtc<DateTime>

Specifies the date and time when the certificate expires.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CertificatePassword<SecureString>

Specifies the password, as a secure string, for the public key infrastructure (PKI) client certificate for the distribution point.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CertificatePath<String>

Specifies the import path for the PKI issued certificate that the distribution point uses.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ClientConnectionType<ClientConnectionTypes>

Specifies the client connection type. Valid values are:

- Internet
- InternetAndIntranet
- Intranet

The acceptable values for this parameter are:

Internet	
InternetAndIntranet	
Intranet	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ComputersUsePxePassword<SecureString>

Specifies the password, as a secure string, for computers that use PXE.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ContentValidationPriority<Priority>

Specifies the content validation priority. Valid values are:

- High
- Highest
- Low

-- Lowest

-- Medium

The default value is Lowest.

The acceptable values for this parameter are:

High	
Highest	
Low	
Lowest	
Medium	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnableAnonymous<Boolean>

Indicates whether the distribution point permits anonymous connections from Configuration Manager clients to the content library.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnablePxeSupport<Boolean>

Indicates whether to enable PXE on the distribution point.

When you enable PXE, Configuration Manager installs Windows Deployment Services on the server, if required. Windows Deployment Service is the service that performs the PXE boot to install operating systems. After you create the distribution point, Configuration Manager installs a provider in Windows Deployment Services that uses the PXE boot functions.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-EnableUnknownComputerSupport<Boolean>

Indicates whether support for unknown computers is enabled. Unknown computers are computers that are not managed by Configuration Manager.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ForceWhenDuplicateCertificate<Boolean>

Indicates whether Configuration Manager overwrites a duplicate certificate when you import a PKI client certificate for the distribution point.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-InitiateConnection<Boolean>

Indicates whether the distribution point initiates the connection with the clients.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InstallationAccount<IResultObject>

Specifies a Site System Installation Account. Configuration Manager 2007 Site Component Manager service uses Site System Installation Accounts to install, reinstall, uninstall, and configure site systems.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InstallInternetServer<Boolean>

Indicates whether Configuration Manager installs and configures Internet Information Services (IIS) on the server if it is not already installed. IIS must be installed on all distribution points.

Aliases	none
Required?	false
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-MacAddressForRespondingPxeRequest<String[]>

Specifies an array of media access controller (MAC) addresses that the distribution point uses to respond to pre-boot execution environment (PXE) requests.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-MinFreeSpaceMB<Int32>

Specifies the amount of free space on a drive before Configuration Manager chooses a different drive and continues the copy process to that drive. Content files can span multiple drives.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PathForSavingMigratedPackage<String>

Specifies the path for a copy of the migrated content.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PrimaryContentLibraryLocation<DriveType>

Specifies the primary content location. Configuration Manager copies content to the primary content location until the amount of free space reaches the value that you specified for the *MinFreeSpaceMB* parameter. Valid values are:

- Automatic.
- Drive letter from A: through Z:.

The acceptable values for this parameter are:

A	
Automatic	
B	
C	
D	
E	
F	
G	
H	
I	
J	
K	
L	
M	
N	
O	
P	
Q	

R	
S	
T	
U	
V	
W	
X	
Y	
Z	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PrimaryPackageShareLocation<DriveType>

Specifies the primary package share location. Configuration Manager copies content to the primary package share location until the amount of free space reaches the value that you specified for the *MinFreeSpaceMB* parameter. Valid values are:

- Automatic.
- Drive letter from A: through Z:.

The acceptable values for this parameter are:

A	
Automatic	
B	
C	
D	
E	

F	
G	
H	
I	
J	
K	
L	
M	
N	
O	
P	
Q	
R	
S	
T	
U	
V	
W	
X	
Y	
Z	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PublicFqdn<String>

Specifies the fully qualified domain name (FQDN) of the site system server that hosts the distribution point.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PxeServerResponseDelaySeconds<Int32>

Specifies, in seconds, how long the distribution point delays before it responds to computer requests when you are using multiple PXE-enabled distribution points. By default, the Configuration Manager PXE service point responds first to network PXE requests.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SecondaryContentLibraryLocation<DriveType>

Specifies the secondary content location. Valid values are:

-- Automatic.

-- Drive letter from A: through Z:.

The acceptable values for this parameter are:

A	
Automatic	
B	

C	
D	
E	
F	
G	
H	
I	
J	
K	
L	
M	
N	
O	
P	
Q	
R	
S	
T	
U	
V	
W	
X	
Y	
Z	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SecondaryPackageShareLocation<DriveType>

Specifies the secondary package share location. Valid values are:

-- Automatic.

-- Drive letter from A: through Z:.

The acceptable values for this parameter are:

A	
Automatic	
B	
C	
D	
E	
F	
G	
H	
I	
J	
K	
L	
M	
N	
O	
P	
Q	
R	
S	
T	
U	
V	
W	
X	

Y	
Z	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SharedDistributionPoint<IResultObject>

Specifies a **CMDistributionPoint** object. To obtain a **CMDistributionPoint** object, use the **Get-CMDistributionPoint** cmdlet.

You can use shared distribution points to make content that you have migrated from a source hierarchy immediately available to clients in the destination hierarchy without having to recreate that content, and then distribute it to new distribution points in the destination hierarchy. When clients in the destination hierarchy request content that is deployed to distribution points that you have shared, the shared distribution points are offered to the clients as valid content locations.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteCode<String>

Specifies the site code for the Configuration Manager site that hosts this site system role.

Aliases	none
Required?	true

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UserDeviceAffinity<UserDeviceAffinityType>

Specify how the distribution point associates users with the destination computer for PXE deployments.

Valid values are:

- AllowWithAutomaticApproval
- AllowWithManualApproval
- DoNotUse

The acceptable values for this parameter are:

AllowWithAutomaticApproval	
AllowWithManualApproval	
DoNotUse	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ValidateContentSchedule<IResultObject>

Specifies a **CMSchedule** object. A **CMSchedule** object defines the schedule for validating the integrity of content files on the distribution point. To create a **CMSchedule** object, use the **New-CMSchedule** cmdlet.

Aliases	none
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Upgrade a shared distribution point

The first command gets the distribution point object that has the ID 6617708D-0F98-4898-8D05-9E882C23DCB2, and stores the object in the \$CIObj variable.

The second command upgrades the shared distribution point stored in \$CIObj to the Configuration Manager site that has the site code CM1. The command specifies the import path for the PKI issued certificate that the distribution point uses, and specifies that the distribution point can pre-stage contents.

```
PS C:\> $CIObj = Get-CMDistributionPoint -DistributionPointGroupId "{6617708D-0F98-4898-8D05-9E882C23DCB2}"
PS C:\> Start-CMDistributionPointUpgrade -AllowPreStaging $True -CertificatePath "\\Contoso01\CM\Toolbox\BaseCert.txt" -SharedDistributionPoint $CIObj -SiteCode "CM1"
```

Related topics

[Get-CMDistributionPointGroup](#)

[Get-CMBoundaryGroup](#)

[Get-CMDistributionPoint](#)

[New-CMSchedule](#)

Start-CMPackageDeployment

Start-CMPackageDeployment

Starts deployment of a software package to a Configuration Manager collection.

Syntax

Parameter Set: DeployDeviceProgramByPackageId

```
Start-CMPackageDeployment -CollectionName <String> -DeviceProgramName <String> -PackageId <String> [-Comment <String> ] [-DeploymentStartDay <DateTime> ] [-DeploymentStartTime <DateTime> ] [-DeployPurpose {Available | Required} ] [-RecurUnit {Days | Hours | Minutes} ] [-RecurValue <Int32> ] [-Rerun <Boolean> ] [-UseMeteredNetwork <Boolean> ] [-UseUtc <Boolean> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: DeployDeviceProgramByPackageName

```
Start-CMPackageDeployment -CollectionName <String> -DeviceProgramName <String> -PackageName <String> [-Comment <String> ] [-DeploymentStartDay <DateTime> ] [-DeploymentStartTime <DateTime> ] [-DeployPurpose {Available | Required} ] [-RecurUnit {Days | Hours | Minutes} ] [-RecurValue <Int32> ] [-Rerun <Boolean> ] [-UseMeteredNetwork <Boolean> ] [-UseUtc <Boolean> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: DeployDeviceProgramByPackageValue

```
Start-CMPackageDeployment -CollectionName <String> -DeviceProgramName <String> -Package <IResultObject> [-Comment <String> ] [-DeploymentStartDay <DateTime> ] [-DeploymentStartTime <DateTime> ] [-DeployPurpose {Available | Required} ] [-RecurUnit {Days | Hours | Minutes} ] [-RecurValue <Int32> ] [-Rerun <Boolean> ] [-UseMeteredNetwork <Boolean> ] [-UseUtc <Boolean> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: DeployStandardProgramByPackageId

```
Start-CMPackageDeployment -CollectionName <String> -PackageId <String> -StandardProgramName <String> [-AllowSharedContent <Boolean> ] [-AllowUsersRunIndependently <Boolean> ] [-Comment <String> ] [-DeploymentAvailableDay <DateTime> ] [-DeploymentAvailableTime <DateTime> ] [-DeploymentExpireDay <DateTime> ] [-DeploymentExpireTime <DateTime> ] [-DeployPurpose {Available | Required} ] [-FastNetworkOption {DownloadContentFromDistributionPointAndRunLocally | RunProgramFromDistributionPoint} ] [-PersistOnWriteFilterDevice <Boolean> ] [-RerunBehavior {AlwaysRetunProgram | NeverRerunDeployedProgram | RerunIfFailedPreviousAttempt | RerunIfSucceededOnpreviousAttempt} ] [-Schedule <IResultObject[]> ] [-ScheduleEvent {AsSoonAsPossible | LogOff | LogOn} ] [-SendWakeUpPacket <Boolean> ] [-SlowNetworkOption {DoNotRunProgram | DownloadContentFromDistributionPointAndLocally | RunProgramFromDistributionPoint} ] [-SoftwareInstallation <Boolean> ] [-SystemRestart <Boolean> ] [-UseMeteredNetwork <Boolean> ] [-UseUtcForAvailableSchedule <Boolean> ] [-UseUtcForExpireSchedule <Boolean> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: DeployStandardProgramByPackageName

```
Start-CMPackageDeployment -CollectionName <String> -PackageName <String> -StandardProgramName <String> [-AllowSharedContent <Boolean> ] [-AllowUsersRunIndependently
```

```
<Boolean> ] [-Comment <String> ] [-DeploymentAvailableDay <DateTime> ] [-
DeploymentAvailableTime <DateTime> ] [-DeploymentExpireDay <DateTime> ] [-
DeploymentExpireTime <DateTime> ] [-DeployPurpose {Available | Required} ] [-
FastNetworkOption {DownloadContentFromDistributionPointAndRunLocally |
RunProgramFromDistributionPoint} ] [-PersistOnWriteFilterDevice <Boolean> ] [-RerunBehavior
{AlwaysRetunProgram | NeverRerunDeployedProgram | RerunIfFailedPreviousAttempt |
RerunIfSucceededOnpreviousAttempt} ] [-Schedule <IResultObject[]> ] [-ScheduleEvent
{AsSoonAsPossible | LogOff | LogOn} ] [-SendWakeUpPacket <Boolean> ] [-SlowNetworkOption
{DoNotRunProgram | DownloadContentFromDistributionPointAndLocally |
RunProgramFromDistributionPoint} ] [-SoftwareInstallation <Boolean> ] [-SystemRestart
<Boolean> ] [-UseMeteredNetwork <Boolean> ] [-UseUtcForAvailableSchedule <Boolean> ] [-
UseUtcForExpireSchedule <Boolean> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

```
Parameter Set: DeployStandardProgramByPackageValue
Start-CMPackageDeployment -CollectionName <String> -Package <IResultObject> -
StandardProgramName <String> [-AllowSharedContent <Boolean> ] [-AllowUsersRunIndependently
<Boolean> ] [-Comment <String> ] [-DeploymentAvailableDay <DateTime> ] [-
DeploymentAvailableTime <DateTime> ] [-DeploymentExpireDay <DateTime> ] [-
DeploymentExpireTime <DateTime> ] [-DeployPurpose {Available | Required} ] [-
FastNetworkOption {DownloadContentFromDistributionPointAndRunLocally |
RunProgramFromDistributionPoint} ] [-PersistOnWriteFilterDevice <Boolean> ] [-RerunBehavior
{AlwaysRetunProgram | NeverRerunDeployedProgram | RerunIfFailedPreviousAttempt |
RerunIfSucceededOnpreviousAttempt} ] [-Schedule <IResultObject[]> ] [-ScheduleEvent
{AsSoonAsPossible | LogOff | LogOn} ] [-SendWakeUpPacket <Boolean> ] [-SlowNetworkOption
{DoNotRunProgram | DownloadContentFromDistributionPointAndLocally |
RunProgramFromDistributionPoint} ] [-SoftwareInstallation <Boolean> ] [-SystemRestart
<Boolean> ] [-UseMeteredNetwork <Boolean> ] [-UseUtcForAvailableSchedule <Boolean> ] [-
UseUtcForExpireSchedule <Boolean> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Start-CMPackageDeployment** cmdlet starts deployment of a specified software package to computers that belong to a Microsoft System Center 2012 Configuration Manager collection. You can choose when the package becomes available and when the package deployment expires. You can specify whether System Center 2012 Configuration Manager deploys the package only once or repeatedly and what happens when installation fails for a computer.

Parameters

-AllowSharedContent<Boolean>

Indicates whether clients use shared content. If this value is \$True, clients attempt to download content from other clients that downloaded that content. If this value is \$False, clients do not attempt to download from other clients.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AllowUsersRunIndependently<Boolean>

Indicates whether users can install the software independently. If this value is \$True, users can install software in this package from the software library regardless of the scheduled installation time. If this value is \$False, the software installs at the scheduled time.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CollectionName<String>

Specifies the ID of a device or user collection.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Comment<String>

Specifies a comment for the deployment.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeploymentAvailableDay<DateTime>

Specifies a day as a **DateTime** object. To obtain a **DateTime** object, use the **Get-Date** cmdlet. For more information, type `Get-Help Get-Date`. This is the day on which the deployment becomes available. If you specify a value for the *DeployAvailableTime* parameter in addition to this parameter, the cmdlet uses that value.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeploymentAvailableTime<DateTime>

Specifies a time as a **DateTime** object. To obtain a **DateTime** object, use the **Get-Date** cmdlet. This is the time at which the deployment becomes available.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeploymentExpireDay<DateTime>

Specifies a day as a **DateTime** object. To obtain a **DateTime** object, use the **Get-Date** cmdlet. This is the day on which the deployment expires. If you specify a value for the *DeploymentExpireTime* parameter in addition to this parameter, the cmdlet uses that value.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeploymentExpireTime<DateTime>

Specifies a time as a **DateTime** object. To obtain a **DateTime** object, use the **Get-Date** cmdlet. This is the time at which the deployment expires.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeploymentStartDay<DateTime>

Specifies a day as a **DateTime** object. To obtain a **DateTime** object, use the **Get-Date** cmdlet. This is the day on which the deployment starts. If you specify a value for the *DeploymentStartTime* parameter in addition to this parameter, the cmdlet uses that value.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeploymentStartTime<DateTime>

Specifies a time as a **DateTime** object. To obtain a **DateTime** object, use the **Get-Date** cmdlet. This is the time at which the deployment starts.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeployPurpose<DeployPurposeType>

Specifies the purpose for the deployment. Valid values are:

- Available
- Required

The acceptable values for this parameter are:

Available	
Required	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeviceProgramName<String>

Specifies the name of a device program.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-FastNetworkOption<FastNetworkOptionType>

Specifies client behavior on a fast network. Valid values are:

- DownloadContentFromDistributionPointAndRunLocally
- RunProgramFromDistributionPoint

The acceptable values for this parameter are:

DownloadContentFromDistributionPointAndRunLocally	
RunProgramFromDistributionPoint	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Package<IResultObject>

Specifies a package object. To obtain a package object, use the **Get-CMPackage** cmdlet.

Aliases	none
---------	------

Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PackageId<String>

Specifies the ID of a package.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PackageName<String>

Specifies the name of a package.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PersistOnWriteFilterDevice<Boolean>

Indicates whether to enable write filters for embedded devices. For a value of \$True, the device commits changes during a maintenance window. This action requires a restart. For a value of \$False, the device saves changes in an overlay and commits them later.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RecurUnit<RecurUnitType>

Specifies a unit for a recurring deployment. Valid values are:

- Days
- Hours
- Minutes

The acceptable values for this parameter are:

Days	
Hours	
Minutes	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RecurValue<Int32>

Specifies how often a deployment recurs. This parameter depends on the unit type specified in the *RecurUnit* parameter. This value can be between 1 and 23 if the unit is Hours, between 1 and 31 if the unit is Days, or between 1 and 59 if the unit is Minutes.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Rerun<Boolean>

Indicates whether the deployment reruns. If this value is \$True, the deployment runs again for clients as specified in the *RerunBehavior* parameter. If this value is \$False, the deployment does not run again.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RerunBehavior<RerunBehaviorType>

Specifies how a deployment reruns on a client. Valid values are:

- AlwaysRetunProgram. Rerun as scheduled, even if the deployment succeeded. You can use this value for recurring deployments.
- NeverRerunDeployedProgram. Does not rerun, even if the deployment failed or files changed.
- RerunIfFailedPreviousAttempt. Rerun, as scheduled, if the deployment failed on the previous attempt.
- RerunIfSucceededOnpreviousAttempt. Rerun only if the previous attempt succeeded. You can use this value for updates that depend on the previous update.

The acceptable values for this parameter are:

AlwaysRetunProgram	
NeverRerunDeployedProgram	
RerunIfFailedPreviousAttempt	
RerunIfSucceededOnpreviousAttempt	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Schedule<IResultObject[]>

Specifies a schedule object for the deployment.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ScheduleEvent<ScheduleEventType[]>

Specifies an array of schedule event types. Valid values are:

- AsSoonAsPossible
- LogOff
- LogOn
- SendWakeUpPacket

The acceptable values for this parameter are:

AsSoonAsPossible	
LogOff	
LogOn	

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SendWakeUpPacket<Boolean>

Indicates whether to send a wake up packet to computers before the deployment begins. If this value is \$True, Configuration Manager wakes a computer from sleep. If this value is \$False, it does not wake computers from sleep. For computers to wake, you must first configure Wake On LAN.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SlowNetworkOption<SlowNetworkOptionType>

Specifies how Configuration Manager deploys this package in a slow network. Valid values are:

- DoNotRunProgram
- DownloadContentFromDistributionPointAndLocally
- RunProgramFromDistributionPoint

The acceptable values for this parameter are:

DoNotRunProgram	
DownloadContentFromDistributionPointAndLocally	
RunProgramFromDistributionPoint	

Aliases	none
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SoftwareInstallation<Boolean>

Indicates whether to install the deployed software outside of maintenance windows. A maintenance window is a specified period of time used for computer maintenance and updates. If this value is \$True, the Configuration Manager installs software according to schedule, even if the schedule falls outside a maintenance window. If this value is \$False, Configuration Manager does not install deployed software outside any windows, but waits for a maintenance window.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-StandardProgramName<String>

Specifies a standard program name.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SystemRestart<Boolean>

Indicates whether a system restarts outside a maintenance window. A maintenance window is a specified period of time used for computer maintenance and updates. If this value is \$True, any

required restart takes place without regard to maintenance windows. If this value is \$False, the computer does not restart outside a maintenance window.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UseMeteredNetwork<Boolean>

Indicates whether to allow clients to download content over a metered Internet connection after the deadline, which may incur additional expense.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UseUtc<Boolean>

Indicates whether to use Coordinated Universal Time (UTC), also known as Greenwich Mean Time. If this value is \$True, Configuration Manager uses UTC for this deployment. If this value is \$False, Configuration Manager uses local time.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UseUtcForAvailableSchedule<Boolean>

Indicates whether to use UTC for available schedule. If this value is \$True, Configuration Manager uses UTC. If this value is \$False, Configuration Manager uses local time.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UseUtcForExpireSchedule<Boolean>

Indicates whether to use UTC for expire schedule. If this value is \$True, Configuration Manager uses UTC. If this value is \$False, Configuration Manager uses local time.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Start a recurring deployment

This command starts deployment for a named package to the collection named All Systems for the device program named DPM. The command specifies a start day and start time. The command includes a descriptive comment. The *Rerun* parameter has a value of \$True and the command specifies a recur value of seven and a recur unit of Days, so deployment reruns every seven days. The deployment uses metered network. The deployment uses UTC time.

```
PS C:\> Start-CMPackageDeployment -CollectionName "All Systems" -DeviceProgramName "DPM" -  
PackageName "User State Migration Tool for Windows 8" -Comment "DPM for all systems." -  
DeploymentStartDay 2012/10/26 -DeploymentStartTime 12:12 -RecurUnit Days -RecurValue 7 -  
Rerun $True -UseMeteredNetwork $True -UseUtc $True
```

Example 2: Start a recurring deployment for an available package

This command starts deployment for a named package to the collection named Western Computers for the device program named DPM. The command includes a descriptive comment. The command specifies Available as the *DeployPurpose*, therefor users can decide whether to install this software. The *Rerun* parameter has a value of \$True. The deployment uses UTC time.

```
PS C:\> Start-CMPackageDeployment -CollectionName "Western Computers" -DeviceProgramName  
"DPM" -PackageName "User State Migration Tool for Windows 8" -Comment "Deployment for  
Western office." -DeployPurpose Available -Rerun $True -UseUtc $True
```

Example 3: Start a deployment for a standard program

This command starts a deployment of a package named User State Migration Tool for Windows 8 to the collection named All Systems for the standard program named SPM. The command does not allow computers to use shared content.

```
PS C:\> Start-CMPackageDeployment -CollectionName "All Systems" -PackageName "User State Migration Tool for Windows 8" -StandardProgramName "SPM" AllowSharedContent $False
```

Related topics

[Set-CMPackageDeployment](#)

[Get-CMPackage](#)

Start-CMSoftwareUpdateDeployment

Start-CMSoftwareUpdateDeployment

Initiates a software update deployment in Configuration Manager.

Syntax

Parameter Set: DeploySoftwareUpdateById

```
Start-CMSoftwareUpdateDeployment -CollectionName <String> -SoftwareUpdateId <String> [-AllowRestart <Boolean> ] [-AllowUseMeteredNetwork <Boolean> ] [-DeploymentAvailableDay <DateTime> ] [-DeploymentAvailableTime <DateTime> ] [-DeploymentExpireDay <DateTime> ] [-DeploymentExpireTime <DateTime> ] [-DeploymentName <String> ] [-DeploymentType {Available | Required} ] [-Description <String> ] [-DisableOperationsManagerAlert <Boolean> ] [-DownloadFromMicrosoftUpdate <Boolean> ] [-GenerateOperationsManagerAlert <Boolean> ] [-GenerateSuccessAlert <Boolean> ] [-PercentSuccess <Int32> ] [-PersistOnWriteFilterDevice <Boolean> ] [-ProtectedType {NoInstall | RemoteDistributionPoint} ] [-RestartServer <Boolean> ] [-RestartWorkstation <Boolean> ] [-SendWakeUpPacket <Boolean> ] [-SoftwareInstallation <Boolean> ] [-TimeBasedOn {LocalTime | UTC} ] [-TimeUnit {Days | Hours | Months | Weeks} ] [-TimeValue <Int32> ] [-UnprotectedType {NoInstall | UnprotectedDistributionPoint} ] [-UseBranchCache <Boolean> ] [-UserNotification {DisplayAll | DisplaySoftwareCenterOnly | HideAll} ] [-VerbosityLevel {AllMessages | OnlyErrorMessages | OnlySuccessAndErrorMessages} ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: DeploySoftwareUpdateByName

```
Start-CMSoftwareUpdateDeployment -CollectionName <String> -SoftwareUpdateName <String> [-AllowRestart <Boolean> ] [-AllowUseMeteredNetwork <Boolean> ] [-DeploymentAvailableDay <DateTime> ] [-DeploymentAvailableTime <DateTime> ] [-DeploymentExpireDay <DateTime> ] [-DeploymentExpireTime <DateTime> ] [-DeploymentName <String> ] [-DeploymentType {Available | Required} ] [-Description <String> ] [-DisableOperationsManagerAlert <Boolean> ] [-DownloadFromMicrosoftUpdate <Boolean> ] [-GenerateOperationsManagerAlert <Boolean> ] [-GenerateSuccessAlert <Boolean> ] [-PercentSuccess <Int32> ] [-PersistOnWriteFilterDevice <Boolean> ] [-ProtectedType {NoInstall | RemoteDistributionPoint} ] [-RestartServer <Boolean> ] [-RestartWorkstation <Boolean> ] [-SendWakeUpPacket <Boolean> ] [-SoftwareInstallation <Boolean> ] [-TimeBasedOn {LocalTime | UTC} ] [-TimeUnit {Days | Hours | Months | Weeks} ] [-TimeValue <Int32> ] [-UnprotectedType {NoInstall | UnprotectedDistributionPoint} ] [-UseBranchCache <Boolean> ] [-UserNotification {DisplayAll | DisplaySoftwareCenterOnly | HideAll} ] [-VerbosityLevel {AllMessages | OnlyErrorMessages | OnlySuccessAndErrorMessages} ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: DeploySoftwareUpdateByValue

```
Start-CMSoftwareUpdateDeployment -CollectionName <String> -SoftwareUpdate <IResultObject> [-AllowRestart <Boolean> ] [-AllowUseMeteredNetwork <Boolean> ] [-DeploymentAvailableDay <DateTime> ] [-DeploymentAvailableTime <DateTime> ] [-DeploymentExpireDay <DateTime> ] [-DeploymentExpireTime <DateTime> ] [-DeploymentName <String> ] [-DeploymentType {Available | Required} ] [-Description <String> ] [-DisableOperationsManagerAlert <Boolean> ] [-DownloadFromMicrosoftUpdate <Boolean> ] [-GenerateOperationsManagerAlert <Boolean> ] [-GenerateSuccessAlert <Boolean> ] [-PercentSuccess <Int32> ] [-PersistOnWriteFilterDevice
```

<Boolean>] [-ProtectedType {NoInstall | RemoteDistributionPoint}] [-RestartServer
<Boolean>] [-RestartWorkstation <Boolean>] [-SendWakeUpPacket <Boolean>] [-
SoftwareInstallation <Boolean>] [-TimeBasedOn {LocalTime | UTC}] [-TimeUnit {Days | Hours
| Months | Weeks}] [-TimeValue <Int32>] [-UnprotectedType {NoInstall |
UnprotectedDistributionPoint}] [-UseBranchCache <Boolean>] [-UserNotification {DisplayAll
| DisplaySoftwareCenterOnly | HideAll}] [-VerbosityLevel {AllMessages | OnlyErrorMessages |
OnlySuccessAndErrorMessages}] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: DeploySoftwareUpdateGroupById

Start-CMSoftwareUpdateDeployment -CollectionName <String> -SoftwareUpdateGroupId <String> [-
AllowRestart <Boolean>] [-AllowUseMeteredNetwork <Boolean>] [-DeploymentAvailableDay
<DateTime>] [-DeploymentAvailableTime <DateTime>] [-DeploymentExpireDay <DateTime>] [-
DeploymentExpireTime <DateTime>] [-DeploymentName <String>] [-DeploymentType {Available |
Required}] [-Description <String>] [-DisableOperationsManagerAlert <Boolean>] [-
DownloadFromMicrosoftUpdate <Boolean>] [-GenerateOperationsManagerAlert <Boolean>] [-
GenerateSuccessAlert <Boolean>] [-PercentSuccess <Int32>] [-PersistOnWriteFilterDevice
<Boolean>] [-ProtectedType {NoInstall | RemoteDistributionPoint}] [-RestartServer
<Boolean>] [-RestartWorkstation <Boolean>] [-SendWakeUpPacket <Boolean>] [-
SoftwareInstallation <Boolean>] [-TimeBasedOn {LocalTime | UTC}] [-TimeUnit {Days | Hours
| Months | Weeks}] [-TimeValue <Int32>] [-UnprotectedType {NoInstall |
UnprotectedDistributionPoint}] [-UseBranchCache <Boolean>] [-UserNotification {DisplayAll
| DisplaySoftwareCenterOnly | HideAll}] [-VerbosityLevel {AllMessages | OnlyErrorMessages |
OnlySuccessAndErrorMessages}] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: DeploySoftwareUpdateGroupName

Start-CMSoftwareUpdateDeployment -CollectionName <String> -SoftwareUpdateGroupName <String>
[-AllowRestart <Boolean>] [-AllowUseMeteredNetwork <Boolean>] [-DeploymentAvailableDay
<DateTime>] [-DeploymentAvailableTime <DateTime>] [-DeploymentExpireDay <DateTime>] [-
DeploymentExpireTime <DateTime>] [-DeploymentName <String>] [-DeploymentType {Available |
Required}] [-Description <String>] [-DisableOperationsManagerAlert <Boolean>] [-
DownloadFromMicrosoftUpdate <Boolean>] [-GenerateOperationsManagerAlert <Boolean>] [-
GenerateSuccessAlert <Boolean>] [-PercentSuccess <Int32>] [-PersistOnWriteFilterDevice
<Boolean>] [-ProtectedType {NoInstall | RemoteDistributionPoint}] [-RestartServer
<Boolean>] [-RestartWorkstation <Boolean>] [-SendWakeUpPacket <Boolean>] [-
SoftwareInstallation <Boolean>] [-TimeBasedOn {LocalTime | UTC}] [-TimeUnit {Days | Hours
| Months | Weeks}] [-TimeValue <Int32>] [-UnprotectedType {NoInstall |
UnprotectedDistributionPoint}] [-UseBranchCache <Boolean>] [-UserNotification {DisplayAll
| DisplaySoftwareCenterOnly | HideAll}] [-VerbosityLevel {AllMessages | OnlyErrorMessages |
OnlySuccessAndErrorMessages}] [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: DeploySoftwareUpdateGroupByValue

Start-CMSoftwareUpdateDeployment -CollectionName <String> -SoftwareUpdateGroup
<IResultObject> [-AllowRestart <Boolean>] [-AllowUseMeteredNetwork <Boolean>] [-
DeploymentAvailableDay <DateTime>] [-DeploymentAvailableTime <DateTime>] [-
DeploymentExpireDay <DateTime>] [-DeploymentExpireTime <DateTime>] [-DeploymentName
<String>] [-DeploymentType {Available | Required}] [-Description <String>] [-
DisableOperationsManagerAlert <Boolean>] [-DownloadFromMicrosoftUpdate <Boolean>] [-
GenerateOperationsManagerAlert <Boolean>] [-GenerateSuccessAlert <Boolean>] [-
PercentSuccess <Int32>] [-PersistOnWriteFilterDevice <Boolean>] [-ProtectedType {NoInstall
| RemoteDistributionPoint}] [-RestartServer <Boolean>] [-RestartWorkstation <Boolean>] [-
SendWakeUpPacket <Boolean>] [-SoftwareInstallation <Boolean>] [-TimeBasedOn {LocalTime |
UTC}] [-TimeUnit {Days | Hours | Months | Weeks}] [-TimeValue <Int32>] [-UnprotectedType

```
{NoInstall | UnprotectedDistributionPoint} ] [-UseBranchCache <Boolean> ] [-UserNotification {DisplayAll | DisplaySoftwareCenterOnly | HideAll} ] [-VerbosityLevel {AllMessages | OnlyErrorMessages | OnlySuccessAndErrorMessages} ] [-Confirm] [-WhatIf] [ <CommonParameters> ]
```

Detailed Description

The **Start-CMSoftwareUpdateDeployment** cmdlet initiates a software update deployment.

Parameters

-AllowRestart<Boolean>

Indicates whether to allow a restart following installation.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AllowUseMeteredNetwork<Boolean>

Indicates whether to allow clients to use a metered network to download updates.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CollectionName<String>

Specifies a name of a collection in Configuration Manager. A collection is a group of client computers.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeploymentAvailableDay<DateTime>

Specifies a day, in MM/DD/YYYY format, when a software update deployment is available. By default, the update is available immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeploymentAvailableTime<DateTime>

Specifies a time, in HH:MM format, when a software update deployment is available. By default, the update is available immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeploymentExpireDay<DateTime>

Specifies a day, in MM/DD/YYYY format, when a software update deployment is no longer available. To expire a software update on a certain day, set this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeploymentExpireTime<DateTime>

Specifies a time, in HH:MM format, when a software update deployment is no longer available. To expire a software update at a certain time, set this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeploymentName<String>

Specifies a name for a software update deployment in Configuration Manager.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-DeploymentType<DeploymentType>

Specifies a deployment type in Configuration Manager.

The acceptable values for this parameter are:

Available	
Required	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Description<String>

Specifies a description for a software update deployment.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DisableOperationsManagerAlert<Boolean>

Indicates whether to disable System Center 2012 – Operations Manager alerts during software updates.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DownloadFromMicrosoftUpdate<Boolean>

Indicates whether clients download updates directly from Microsoft Update.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-GenerateOperationsManagerAlert<Boolean>

Indicates whether to generate Operations Manager alerts when a software installation fails.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-GenerateSuccessAlert<Boolean>

Indicates whether to generate alerts when a software installation succeeds.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PercentSuccess<Int32>

Specifies a percent success.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PersistOnWriteFilterDevice<Boolean>

Indicates whether to install a software update on the temporary overlay and commit changes later, or commit the changes at an installation deadline or a maintenance window.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ProtectedType<ProtectedType>

Specifies a protected type.

The acceptable values for this parameter are:

NoInstall	
RemoteDistributionPoint	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RestartServer<Boolean>

Indicates whether to allow a server to restart following a software update. Setting this value to \$True prevents the server from restarting. Setting this value to \$False allows the server to restart.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-RestartWorkstation<Boolean>

Indicates whether to allow a workstation to restart following a software update. Setting this value to \$True prevents the computer from restarting. Setting this value to \$False allows the computer to restart.

Aliases	none
Required?	false
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SendWakeUpPacket<Boolean>

Indicates whether to send a wake up packet to computers before the deployment begins. If this value is \$True, Configuration Manager wakes a computer from sleep. If this value is \$False, it does not wake computers from sleep. For computers to wake, you must first configure Wake On LAN.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SoftwareInstallation<Boolean>

Indicates whether to allow the software update to install, even if the installation occurs outside of a maintenance window.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SoftwareUpdate<IResultObject>

Specifies a software update deployment object. To obtain a software update object, use the [Get-CMSoftwareUpdate](#) cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SoftwareUpdateGroup<IResultObject>

Specifies a software update group object. To obtain a software update group object, use the [Get-CMSoftwareUpdateGroup](#) cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SoftwareUpdateGroupId<String>

Specifies an ID for a software update group. A software update group contains individual software updates.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SoftwareUpdateGroupName<String>

Specifies a name for a software update group.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SoftwareUpdateId<String>

Specifies an ID for a software update in Configuration Manager.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SoftwareUpdateName<String>

Specifies a name for a software update in Configuration Manager.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-TimeBasedOn<TimeType>

Specifies that client computers use either local or UTC time to determine the availability of a program. UTC time makes the software update available at the same time for all computers.

The acceptable values for this parameter are:

LocalTime	
UTC	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-TimeUnit<TimeUnitType>

Specifies the time unit in Configuration Manager. Valid values are:

- Days
- Hours
- Months
- Weeks

The acceptable values for this parameter are:

Days	
Hours	
Months	
Weeks	

Aliases	none
Required?	false
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-TimeValue<Int32>

Specifies a time value in the units specified in the *TimeUnit* parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UnprotectedType<UnprotectedType>

Specifies an unprotected type.

The acceptable values for this parameter are:

NoInstall	
UnprotectedDistributionPoint	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UseBranchCache<Boolean>

Indicates whether to use Branch Cache as a distribution point for updates.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UserNotification<UserNotificationType>

Specifies a user notification type.

The acceptable values for this parameter are:

DisplayAll	
DisplaySoftwareCenterOnly	
HideAll	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-VerbosityLevel<VerbosityLevelType>

Specifies verbosity level. Valid values are:

- AllMessages
- OnlyErrorMessage
- OnlySuccessAndErrorMessage

The acceptable values for this parameter are:

AllMessages	
OnlyErrorMessage	
OnlySuccessAndErrorMessage	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Start a required deployment by software update name

This command starts a required software update deployment by using a software update name.

```
PS C:\> Start-CMSoftwareUpdateDeployment -SoftwareUpdateName "CT" -CollectionName "All Systems" -DeploymentName "Contoso-test" -Description "Contoso-test-deployment" -DeploymentType Required -SendWakeUpPacket $True -VerbosityLevel AllMessages -TimeBasedOn UTC -DeploymentAvailableDay 2012/10/24 -DeploymentAvailableTime 23:56 -DeploymentExpireDay 2013/10/21 -DeploymentExpireTime 11:20 -UserNotification HideAll -SoftwareInstallation $True -AllowRestart $True -RestartServer $True -RestartWorkstation $True -PersistOnWriteFilterDevice $False -GenerateSuccessAlert $True -PercentSuccess 90 -TimeValue 10 -TimeUnit Days -DisableOperationsManagerAlert $True -GenerateOperationsManagerAlert $True -ProtectedType RemoteDistributionPoint -UnprotectedType NoInstall -UseBranchCache $False -DownloadFromMicrosoftUpdate $True -AllowUseMeteredNetwork $True
```

Example 2: Start an available deployment by software update name

This command starts an available software update deployment by using a software update name.

```
PS C:\> Start-CMSoftwareUpdateDeployment -SoftwareUpdateName "CT" -CollectionName "All Systems" -DeploymentName "Contoso-test2" -Description "Contoso-test2-deployment" -DeploymentType Available -VerbosityLevel AllMessages -TimeBasedOn UTC -DeploymentAvailableDay 2012/10/24 -DeploymentAvailableTime 23:56 -UserNotification DisplayAll -PersistOnWriteFilterDevice $False -DisableOperationsManagerAlert $True -GenerateOperationsManagerAlert $True -ProtectedType RemoteDistributionPoint -UnprotectedType NoInstall -UseBranchCache $False -DownloadFromMicrosoftUpdate $True -AllowUseMeteredNetwork $True
```

Example 3: Start a required deployment by software update group name

This command starts a software update deployment by using a collection name and an input object.

```
PS C:\> Start-CMSoftwareUpdateDeployment -SoftwareUpdateGroupName "CTG" -CollectionName "All Systems" -DeploymentName "Contoso-test3" -Description "Contoso-test3-deployment" -DeploymentType Required -SendWakeUpPacket $True -VerbosityLevel AllMessages -TimeBasedOn UTC -DeploymentAvailableDay 2012/10/24 -DeploymentAvailableTime 23:56 -DeploymentExpireDay 2013/10/21 -DeploymentExpireTime 11:20 -UserNotification HideAll -SoftwareInstallation $True -AllowRestart $True -RestartServer $True -RestartWorkstation $True -PersistOnWriteFilterDevice $False -GenerateSuccessAlert $True -PercentSuccess 90 -TimeValue 10 -TimeUnit Days -DisableOperationsManagerAlert $True -GenerateOperationsManagerAlert $True
```

```
-ProtectedType RemoteDistributionPoint -UnprotectedType NoInstall -UseBranchCache $False -  
DownloadFromMicrosoftUpdate $True -AllowUseMeteredNetwork $True
```

Example 4: Start a deployment by software update group name

This command starts a software update deployment by using a software update group name.

```
PS C:\> Start-CMSoftwareUpdateDeployment -SoftwareUpdateGroupName "CTG" -CollectionName "All  
Systems" -DeploymentName "Contoso-test4" -Description "Contoso-test4-deployment" -  
DeploymentType Available -VerbosityLevel AllMessages -TimeBasedOn UTC -  
DeploymentAvailableDay 2012/10/24 -DeploymentAvailableTime 23:56 -UserNotification  
DisplayAll -PersistOnWriteFilterDevice $False -DisableOperationsManagerAlert $True -  
GenerateOperationsManagerAlert $True -ProtectedType RemoteDistributionPoint -UnprotectedType  
NoInstall -UseBranchCache $False -DownloadFromMicrosoftUpdate $True -AllowUseMeteredNetwork  
$True
```

Related topics

[Set-CMSoftwareUpdateDeployment](#)

Start-CMTaskSequenceDeployment

Start-CMTaskSequenceDeployment

Starts a task sequence deployment in Configuration Manager.

Syntax

Parameter Set: SearchPackageByIdMandatory

```
Start-CMTaskSequenceDeployment -CollectionName <String> -TaskSequencePackageId <String[]> [-AlertDay <DateTime> ] [-AlertTime <DateTime> ] [-AllowFallback <Boolean> ] [-AllowSharedContent <Boolean> ] [-AllowUsersRunIndependently <Boolean> ] [-Comment <String> ] [-CreateAlertBaseOnPercentFailure <Boolean> ] [-CreateAlertBaseOnPercentSuccess <Boolean> ] [-DeploymentAvailableDay <DateTime> ] [-DeploymentAvailableTime <DateTime> ] [-DeploymentExpireDay <DateTime> ] [-DeploymentExpireTime <DateTime> ] [-DeploymentOption {DownloadAllContentLocallyBeforeStartingTaskSequence | DownloadContentLocallyWhenNeededByRunningTaskSequence} ] [-DeployPurpose {Available | Required} ] [-InternetOption <Boolean> ] [-MakeAvailableTo <MakeAvailableToType> ] [-PercentFailure <Int32> ] [-PercentSuccess <Int32> ] [-PersistOnWriteFilterDevice <Boolean> ] [-RerunBehavior {AlwaysRerunProgram | NeverRerunDeployedProgram | RerunIfFailedPreviousAttempt | RerunIfSucceededOnPreviousAttempt} ] [-Schedule <IResultObject[]> ] [-ScheduleEvent {AsSoonAsPossible | LogOff | LogOn} ] [-SendWakeUpPacket <Boolean> ] [-ShowTaskSequenceProgress <Boolean> ] [-SoftwareInstallation <Boolean> ] [-SystemRestart <Boolean> ] [-UseMeteredNetwork <Boolean> ] [-UseUtcForAvailableSchedule <Boolean> ] [-UseUtcForExpireSchedule <Boolean> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Start-CMTaskSequenceDeployment -CollectionName <String> -InputObject <IResultObject> [-AlertDay <DateTime> ] [-AlertTime <DateTime> ] [-AllowFallback <Boolean> ] [-AllowSharedContent <Boolean> ] [-AllowUsersRunIndependently <Boolean> ] [-Comment <String> ] [-CreateAlertBaseOnPercentFailure <Boolean> ] [-CreateAlertBaseOnPercentSuccess <Boolean> ] [-DeploymentAvailableDay <DateTime> ] [-DeploymentAvailableTime <DateTime> ] [-DeploymentExpireDay <DateTime> ] [-DeploymentExpireTime <DateTime> ] [-DeploymentOption {DownloadAllContentLocallyBeforeStartingTaskSequence | DownloadContentLocallyWhenNeededByRunningTaskSequence} ] [-DeployPurpose {Available | Required} ] [-InternetOption <Boolean> ] [-MakeAvailableTo <MakeAvailableToType> ] [-PercentFailure <Int32> ] [-PercentSuccess <Int32> ] [-PersistOnWriteFilterDevice <Boolean> ] [-RerunBehavior {AlwaysRerunProgram | NeverRerunDeployedProgram | RerunIfFailedPreviousAttempt | RerunIfSucceededOnPreviousAttempt} ] [-Schedule <IResultObject[]> ] [-ScheduleEvent {AsSoonAsPossible | LogOff | LogOn} ] [-SendWakeUpPacket <Boolean> ] [-ShowTaskSequenceProgress <Boolean> ] [-SoftwareInstallation <Boolean> ] [-SystemRestart <Boolean> ] [-UseMeteredNetwork <Boolean> ] [-UseUtcForAvailableSchedule <Boolean> ] [-UseUtcForExpireSchedule <Boolean> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Start-CMTaskSequenceDeployment** cmdlet starts a task sequence deployment. A task sequence deployment assigns a task sequence to a collection of computers.

Parameters

-AlertDay<DateTime>

Specifies a day, in MM/DD/YYYY format, to notify clients of a new deployment.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AlertTime<DateTime>

Specifies a time, in HH:MM format, to notify clients of a new deployment.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AllowFallback<Boolean>

Indicates whether to allow a fallback status point for clients.

Aliases	none
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AllowSharedContent<Boolean>

Indicates whether to allow shared content, such as a shared network folder.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-AllowUsersRunIndependently<Boolean>

Indicates whether to allow users to independently run the program, regardless of its assignment status.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CollectionName<String>

Specifies a name of a collection designated to receive a task sequence deployment. A collection is a group of client computers.

Aliases	none
---------	------

Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Comment<String>

Specifies a comment for the task sequence deployment.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CreateAlertBaseOnPercentFailure<Boolean>

Indicates whether to create an alert, depending on the percentage of an installation that fails. If you specify a percent failure for installations, Configuration Manager raises an alert in the console when deployment fails on that percentage of installations. Use this parameter in conjunction with the *PercentFailure* parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CreateAlertBaseOnPercentSuccess<Boolean>

Indicates whether to create an alert, depending on the percentage of an installation that succeeds. If you specify a percent success for installations, Configuration Manager raises an alert in the console when the deployment succeeds on that percentage of installations. Use this parameter in conjunction with the *PercentSuccess* parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeploymentAvailableDay<DateTime>

Specifies a day, in MM/DD/YYYY format, when a deployment becomes available to clients. By default, the deployment becomes available immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeploymentAvailableTime<DateTime>

Specifies a time, in HH:MM format, when a deployment becomes available to clients. By default, the deployment becomes available immediately.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeploymentExpireDay<DateTime>

Specifies a day, in MM/DD/YYYY format, when a deployment expires. By default, a deployment never expires. To have a deployment expire on a certain day, set this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeploymentExpireTime<DateTime>

Specifies a time, in HH:MM format, when the deployment expires. By default, a deployment never expires. To have a deployment expire at a certain time, set this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeploymentOption<DeploymentOptionType>

Specifies if clients download all content before starting the task sequence, or download content as needed by the running task sequence. By default, clients download content as needed. Valid values are:

- DownloadAllContentLocallyBeforeStartingTaskSequence
- DownloadContentLocallyWhenNeededByRunningTaskSequence

The acceptable values for this parameter are:

DownloadAllContentLocallyBeforeStartingTaskSequence	
DownloadContentLocallyWhenNeededByRunningTaskSequence	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeployPurpose<DeployPurposeType>

Specifies a task sequence as either required or available. A required task sequence installation is mandatory. Valid values are:

- Available
- Required

The acceptable values for this parameter are:

Available	
Required	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a task sequence deployment object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InternetOption<Boolean>

Indicates whether the task sequence runs on clients connecting over the Internet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-MakeAvailableTo<MakeAvailableToType>

Specifies the collections that receive this deployment. Valid values are:

- Clients
- ClientsMediaAndPxe
- MediaAndPxe
- MediaAndPxeHidden

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PercentFailure<Int32>

Specifies a percentage for failed task sequence deployment.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PercentSuccess<Int32>

Specifies a percentage for successful task sequence deployment.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PersistOnWriteFilterDevice<Boolean>

Indicates whether to install a task sequence on the temporary overlay and commit changes later, or commit the changes at an installation deadline or a maintenance window.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-RerunBehavior<RerunBehaviorType>

Specifies that a task sequence will be rerun on a computer if it has previously been run before the scheduled mandatory time. By default, the task sequence is always rerun. Valid values are:

- AlwaysRerunProgram
- NeverRerunDeployedProgram
- RerunIfFailedPreviousAttempt
- RerunIfSucceededOnPreviousAttempt

The acceptable values for this parameter are:

AlwaysRerunProgram	
NeverRerunDeployedProgram	
RerunIfFailedPreviousAttempt	
RerunIfSucceededOnPreviousAttempt	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Schedule<IResultObject[]>

Specifies an array of schedule objects.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-ScheduleEvent<ScheduleEventType[]>

Specifies an array of events that determine when the task sequence deployment runs. Valid values are:

- AsSoonAsPossible
- LogOff
- LogOn

The acceptable values for this parameter are:

AsSoonAsPossible	
LogOff	
LogOn	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SendWakeUpPacket<Boolean>

Indicates whether to send a wake up packet to computers before the deployment begins. If this value is \$True, Configuration Manager wakes a computer from sleep. If this value is \$False, it does not wake computers from sleep. For computers to wake, you must first configure Wake On LAN. The *Purpose* parameter must be set to Required.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-ShowTaskSequenceProgress<Boolean>

Indicates whether to show a process dialog for a task sequence.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SoftwareInstallation<Boolean>

Indicates whether to allow the application to install, even if the installation occurs outside of a maintenance window.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SystemRestart<Boolean>

Indicates whether to allow an advertised program to restart the system, even if the restart occurs outside of a maintenance window.

Aliases	none
Required?	false
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-TaskSequencePackageId<String[]>

Specifies an array of IDs for a task sequence package.

Aliases	PackageId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UseMeteredNetwork<Boolean>

Indicates whether to allow clients to connect to a metered network to download a program.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UseUtcForAvailableSchedule<Boolean>

Indicates whether client computers use UTC time to determine the availability of a program. UTC time makes the task sequence available at the same time for all computers.

Aliases	none
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-UseUtcForExpireSchedule<Boolean>

Indicates whether client computers use UTC time to determine the expiration of a program. UTC time makes the task sequence available at the same time for all computers.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Start a task sequence deployment

This command starts a task sequence deployment by using the name of the task sequence deployment and the name of a collection.

```
PS C:\> Start-CMTaskSequenceDeployment -Name "Task Sequence 1333" -CollectionName "All Systems"
```

Example 2: Start a task sequence deployment for devices

This command starts a task sequence deployment for mobile devices. The command does not allow the use of the *PersistOnWriteFilterDevice* parameter.

```
PS C:\> Start-CMTaskSequenceDeployment -Name "Task Sequence 1333" -CollectionName "All Mobile Devices" -Comment "Task sequence test" -DeployPurpose Required -SendWakeUpPacket $True -UseMeteredNetwork $True -ScheduleEvent AsSoonAsPossible -RerunBehavior NeverRerunDeployedProgram -AllowUsersRunIndependently $True -ShowTaskSequenceProgress $False -SoftwareInstallation $True -SystemRestart $True -PersistOnWriteFilterDevice $False -AllowFallback $True -CreatAlertBaseOnPercentSuccess $True -CreatAlertBaseOnPercentFailure $True -DeploymentOption DownloadAllContentLocallyBeforeStartingTaskSequence -AllowSharedContent $True -InternetOption $True
```

Related topics

[Set-CMTaskSequenceDeployment](#)

Stop-CMCloudDistributionPoint

Stop-CMCloudDistributionPoint

Stops the cloud distribution point service.

Syntax

Parameter Set: SearchByIdMandatory

```
Stop-CMCloudDistributionPoint -Id <String[]> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByNameMandatory

```
Stop-CMCloudDistributionPoint -Name <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Stop-CMCloudDistributionPoint -InputObject <IResultObject> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Stop-CMCloudDistributionPoint** cmdlet stops the cloud distribution point service.

If you use the **Stop-CMCloudDistributionPoint** cmdlet, System Center 2012 Configuration Manager does not delete content from the distribution point and does not prevent the site server from transferring additional content to the distribution point. While the cloud distribution point service is stopped, the cloud distribution point does not distribute content. Use the **Start-CMCloudDistributionPoint** cmdlet to restart distribution.

For example, you might want to stop a cloud service when usage reaches a data threshold and then restart it at a later time.

Parameters

-Id<String[]>

Specifies an array of identifiers for cloud distribution points. You can use a comma separated list.

Aliases	AzureServiceId
Required?	true
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a cloud distribution point object. To obtain a cloud distribution point object, you can use the **Get-CMCloudDistributionPoint** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies the name of a cloud distribution point.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Stop the cloud distribution point service using an ID

This command stops the cloud distribution point service for the cloud distribution point that has the specified identifier.

```
PS C:\> Stop-CMCloudDistributionPoint -Id "16777242"
```

Example 2: Stop the cloud distribution point service using a name

This command stops the cloud distribution point service for the cloud distribution point named West01.

```
PS C:\> Stop-CMCloudDistributionPoint -Name "West01"
```

Example 3: Stop the cloud distribution point service using an object

The first command uses the **Get-CMCloudDistributionPoint** cmdlet to get the distribution point with the specified identifier, and then stores it in the \$DistPnt variable.

The second command stops the cloud distribution point service for the distribution point stored in \$DistPnt.

```
PS C:\> $DistPnt = Get-CMCloudDistributionPoint -Id "16777242"  
PS C:\> Stop-CMCloudDistributionPoint -InputObject $DistPnt
```

Related topics

[Get-CMCloudDistributionPoint](#)

[New-CMCloudDistributionPoint](#)

[Remove-CMCloudDistributionPoint](#)

[Set-CMCloudDistributionPoint](#)

[Start-CMCloudDistributionPoint](#)

Suspend-CMAAlert

Suspend-CMAAlert

Suspends monitoring alerts.

Syntax

Parameter Set: SearchByIdMandatory

```
Suspend-CMAAlert -Id <String> -SkipUntil <DateTime> [-Comment <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByNameMandatory

```
Suspend-CMAAlert -Name <String> -SkipUntil <DateTime> [-Comment <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Suspend-CMAAlert -InputObject <IResultObject> -SkipUntil <DateTime> [-Comment <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Suspend-CMAAlert** cmdlet suspends monitoring of an alert until a specified date. At that time, Microsoft System Center 2012 Configuration Manager updates the state of the alert. You can suspend an alert only when it is enabled. If you do not specify the *SkipUntil* parameter, the alert is suspended indefinitely.

Parameters

-Comment<String>

Specifies a comment to add to the alert. You can use the comment to record the explanation for suspending the alert.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Id<String>

Specifies an alert identifier. You can obtain the identifier of an alert by using the **Get-Alert** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a **CMAAlert** object. To obtain a **CMAAlert** object, use the **Get-CMAAlert** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String>

Specifies the name of an alert. You can obtain the name of an alert by using **Get-CMAAlert**.

Aliases	none
Required?	true
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SkipUntil<DateTime>

Specifies a specific date and time to start evaluation of the alert. Enter a **DateTime** object or a string that can be converted to a time, such as April 19, 2012 15:00, 12/31/2013 9:00 PM, or 3am. To obtain a **DateTime** object, use the **Get-Date** cmdlet. For more information, type `Get-Help Get-Date`.

If you do not specify an element of the **DateTime** object, such as seconds, that element of the job trigger is not changed. If the original job trigger did not include a **DateTime** object and you omit an element, the job trigger is created with the corresponding element from the current date and time.

DateTime objects, and strings that are converted to **DateTime** objects, are automatically adjusted to be compatible with the date and time formats selected for the local computer in **Region and Language** in Control Panel.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Suspend an alert by using ID

This command suspends an alert that has the Id 16777219 until the time specified by *SkipUntil*, and adds a comment to the alert.

```
PS C:\> Suspend-CMAAlert -Id "16777219" -Comments "Postponing alert evaluation" -SkipUntil "Wednesday, August 20, 2012 4:03:17 PM"
```

Example 2: Suspend an alert by using alert object variable

The first command gets the alert object that has the Id 16777221 and stores the object in the \$AlertObj variable.

The second command suspends the alert stored in \$AlertObj until the time specified by *SkipUntil*, and adds a comment to the alert.

```
PS C:\> $AlertObj = Get-CMAAlert -Id "16777221"
PS C:\> Suspend-CMAAlert -InputObject $AlertObj -Comments "Postponing alert evaluation" -SkipUntil "4/8/2012 8:04:39 PM"
```

Related topics

[Enable-CMAAlert](#)

[Get-CMAAlert](#)

[Remove-CMAAlert](#)

[Set-CMAAlert](#)

[Disable-CMAAlert](#)

Suspend-CMApplication

Suspend-CMApplication

Suspends an application in Configuration Manager.

Syntax

Parameter Set: SearchByIdMandatory

```
Suspend-CMApplication -Id <String[]> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByNameMandatory

```
Suspend-CMApplication -Name <String[]> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Suspend-CMApplication -InputObject <IResultObject> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Suspend-CMApplication** cmdlet suspends an application. Until the application is resumed, users cannot modify or deploy the application. This action does not affect existing deployments.

Parameters

-Id<String[]>

Specifies an array of application IDs.

Aliases	CIId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies an application object. To obtain an application object, use the **Get-CMApplication** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String[]>

Specifies an array of application names.

Aliases	LocalizedDisplayName
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Suspend an application by using ID

This command suspends an application by ID.

```
PS C:\> Suspend-CMApplication -Id "16781415"
```

Related topics

[Export-CMApplication](#)

[Get-CMApplication](#)

[Import-CMApplication](#)

[New-CMApplication](#)

[Remove-CMApplication](#)

[Resume-CMApplication](#)

[Set-CMApplication](#)

Sync-CMAssetIntelligenceCatalog

Sync-CMAssetIntelligenceCatalog

Synchronizes the Asset Intelligence catalog with System Center Online.

Syntax

```
Sync-CMAssetIntelligenceCatalog [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Sync-CMAssetIntelligenceCatalog** cmdlet synchronizes the local Asset Intelligence catalog with System Center Online to retrieve the latest software title categorization. The Asset Intelligence catalog contains categorization and identification information for software titles. System Center Online accepts only one manual synchronization request in a 12-hour period.

Microsoft System Center 2012 Configuration Manager updates the asset Intelligence catalog by using the Asset Intelligence synchronization point site system role. You must install an Asset Intelligence synchronization point site system role before you can synchronize the Asset Intelligence catalog with System Center Online. You can use the [Add-CMAssetIntelligenceSynchronizationPoint](#) cmdlet to install an Asset Intelligence synchronization point site system role.

When you manually request catalog synchronization with System Center Online, it could take 15 minutes or longer to complete the synchronization process with System Center Online.

Parameters

-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

Example 1: Update the Asset Intelligence catalog

This command updates the Asset Intelligence catalog on the System Center 2012 Configuration Manager site that has the site code CM2 on the site system server named Contoso-west02.

```
PS C:\> Sync-CMAAssetIntelligenceCatalog -SiteCode "CM2" -SiteSystemServerName "Contoso-west02"
```

Related topics

[Get-CMAAssetIntelligenceSynchronizationPoint](#)

[Set-CMAAssetIntelligenceSynchronizationPoint](#)

[Add-CMAAssetIntelligenceSynchronizationPoint](#)

[Send-CMAAssetIntelligenceCatalogUpdateRequest](#)

Sync-CMExchangeServer

Sync-CMExchangeServer

Synchronizes Configuration Manager mobile device information with an Exchange Server.

Syntax

Parameter Set: SearchBySiteCodeMandatory

```
Sync-CMExchangeServer -Address <String> -SiteCode <String> [-Force] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Detailed Description

The **Sync-CMExchangeServer** cmdlet synchronizes mobile device information with a specified Microsoft Exchange Server for one or more System Center 2012 Configuration Manager sites.

System Center 2012 Configuration Manager works with Exchange Server to manage mobile devices that cannot run System Center 2012 Configuration Manager clients.

Parameters

-Address<String>

Specifies a URL for the Exchange Server.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SiteCode<String>

Specifies the site code for a Configuration Manager site associated with the Exchange Server.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Synchronize with an Exchange Server

This command synchronizes mobile devices for the site that has the site code PE1 with the specified Exchange Server.

```
PS C:\> Sync-CMExchangeServer -Address "http://localhost/PowerShell" -SiteCode "PE1"
```

Related topics

[Get-CMExchangeServer](#)

[New-CMExchangeServer](#)

[Remove-CMExchangeServer](#)

[Set-CMExchangeServer](#)

Sync-CMSoftwareUpdate

Sync-CMSoftwareUpdate

Retrieves metadata for software updates.

Syntax

Parameter Set: Default

```
Sync-CMSoftwareUpdate -FullSync <Boolean> [ <CommonParameters>]
```

Detailed Description

The **Sync-CMSoftwareUpdate** cmdlet retrieves metadata for software updates. You can use this cmdlet to retrieve metadata for all software updates or only recent changes to software updates. Software update synchronization in Microsoft System Center 2012 Configuration Manager is the process of retrieving the software updates metadata that meets the criteria that you configure.

Parameters

-FullSync<Boolean>

Indicates whether to perform a complete synchronization of all updates or a delta synchronization.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

<CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

Examples

Example 1: Perform a full synchronization of software updates

This command retrieves metadata for all software updates.

```
PS C:\> Sync-CMSoftwareUpdate -FullSync $True
```

Related topics

[Get-CMSoftwareUpdate](#)

[Save-CMSoftwareUpdate](#)

[Set-CMSoftwareUpdate](#)

Unblock-CMCertificate

Unblock-CMCertificate

Unblocks certificates.

Syntax

Parameter Set: SearchByIdMandatory

```
Unblock-CMCertificate -Id <String[]> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Unblock-CMCertificate** cmdlet unblocks one or more public key infrastructure (PKI) certificates that Microsoft System Center 2012 Configuration Manager uses.

Parameters

-Id<String[]>

Specifies an array of IDs of certificates.

Aliases	Smsid
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Unblock a certificate

This command unblocks the PKI certificate named BaseCert.

```
PS C:\> Unblock-CMCertificate -Id "BaseCert.txt"
```

Related topics

[Block-CMCertificate](#)

[Import-CMCertificate](#)

[Update-CMCertificate](#)

Unblock-CMDevice

Unblock-CMDevice

Unblocks Configuration Manager client devices.

Syntax

Parameter Set: SearchByNameMandatory

```
Unblock-CMDevice -DeviceName <String[]> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Unblock-CMDevice -DeviceId <String[]> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Unblock-CMDevice -InputObject <IResultObject> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Unblock-CMDevice** cmdlet unblocks one or more Microsoft System Center 2012 Configuration Manager client devices. You must unblock a device from the client's assigned site. You cannot unblock the device from sites higher in the hierarchy.

If you unblock an Intel Active Management Technology (Intel AMT)-based computer that you provisioned for AMT when it was blocked, you must take additional steps before you can manage that computer again out of band.

For more information about unblocking a client device in System Center 2012 Configuration Manager, see [Determine Whether to Block Clients in Configuration Manager](http://go.microsoft.com/fwlink/?LinkId=271116) (<http://go.microsoft.com/fwlink/?LinkId=271116>) on TechNet.

Parameters

-DeviceId<String[]>

Specifies an array of device IDs.

Aliases	ResourceId
Required?	true
Position?	named

Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeviceName<String[]>

Specifies an array of device names.

Aliases	Name
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a **CMDevice** object. To obtain a **CMDevice** object, use the [Get-CMDevice](#) cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Unblock a client device

This command unblocks the client device named CMCEN-DIST02.

```
PS C:\> Unblock-CMDevice -DeviceName "CMCEN-DIST02"
```

Related topics

[Get-CMDevice](#)

[Approve-CMDevice](#)

[Block-CMDevice](#)

[Remove-CMDevice](#)

Unblock-CMThreat

Unblock-CMThreat

Unblocks a threat.

Syntax

Parameter Set: SearchByNameMandatoryForAllowThreat

```
Unblock-CMThreat -CollectionName <String> -ThreatName <String> [-Force] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByIdMandatoryForAllowThreat

```
Unblock-CMThreat -CollectionId <String> -ThreatId <String> [-Force] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByIdMandatoryForRestoreFiles

```
Unblock-CMThreat -ApplyAMPolicy <Boolean> -CollectionId <String> -ThreatId <String> [-Force] [  
-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByNameMandatoryForRestoreFiles

```
Unblock-CMThreat -ApplyAMPolicy <Boolean> -CollectionName <String> -ThreatName <String> [-  
Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatoryForAllowThreat

```
Unblock-CMThreat -InputObject <IResultObject> [-Force] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByValueMandatoryForRestoreFiles

```
Unblock-CMThreat -ApplyAMPolicy <Boolean> -InputObject <IResultObject> [-Force] [-Confirm] [  
-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Unblock-CMThreat** cmdlet unblocks a threat.

Parameters

-ApplyAMPolicy<Boolean>

Indicates whether to apply anti-malware policy.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CollectionId<String>

Specifies a collection ID associated with a threat.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-CollectionName<String>

Specifies a name of a collection associated with a threat.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a **CMThreat** object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ThreatId<String>

Specifies an ID of a threat.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ThreatName<String>

Specifies a threat name.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Unblock a threat

This command unblocks a threat that Configuration Manager identifies by collection name and threat name.

```
PS C:\> Unblock-CMThreat -CollectionName "Coll01" -ThreatName "Threat01"
```

Undo-CMSoftwareInventory

Undo-CMSoftwareInventory

Stops collecting software inventory data on files.

Syntax

Parameter Set: SearchByNameMandatory

```
Undo-CMSoftwareInventory -Name <String[]> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Undo-CMSoftwareInventory -Id <String[]> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Undo-CMSoftwareInventory -SoftwareInventory <IResultObject> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Undo-CMSoftwareInventory** cmdlet stops collecting information about files that are contained on client devices.

Parameters

-Id<String[]>

Specifies an array of IDs of software files.

Aliases	SoftwareKey
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String[]>

Specifies an array of names of software files.

Aliases	CommonName
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	true

-SoftwareInventory<IResultObject>

Specifies a **CMSSoftwareInventory** object. To obtain a **CMSSoftwareInventory** object, use the [Get-**CMSSoftwareInventory**](#) cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Stop collecting software inventory data on a file

This command stops collecting software inventory data on the file named MSXML 6.0 Parser.

```
PS C:\> Undo-CMSoftwareInventory -Name "MSXML 6.0 Parser"
```

Related topics

[Get-CMSoftwareInventory](#)

[Set-CMSoftwareInventory](#)

Unlock-CMObject

Unlock-CMObject

Releases locks to global objects in Configuration Manager.

Syntax

Parameter Set: ByValue

```
Unlock-CMObject [-InputObject] <IResultObject[]> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Unlock-CMObject** cmdlet releases locks of one or more objects in Microsoft System Center 2012 Configuration Manager. You can use the *InputObject* parameter to specify the input to this cmdlet, or you can pipe the input to this cmdlet.

Serialized Editing of Distributed Objects (SEDO) in System Center 2012 Configuration Manager provide a mechanism for assigning and unassigning locks to global System Center 2012 Configuration Manager objects in the context of a site, computer and user. If you want to edit and save a SEDO-enabled object, you must lock the object. When you obtain the lock, the lock is assigned to you, your computer and the site in which the computer resides. While the lock is assigned to you, no other user or computer can edit the object until you release the lock.

Parameters

-InputObject<IResultObject[]>

Specifies an array of Configuration Manager objects output from another cmdlet.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Unlock a global object

The first command gets the driver package object that has the ID CM100042, and stores the object in the \$CIObj variable.

The second command releases the lock the object stored in \$CIObj.

```
PS C:\> $CIObj = Get-CMDriverPackage -Id "CM100042"  
PS C:\> Lock-CMObject $CIObj
```

Related topics

[Lock-CMObject](#)

[Move-CMObject](#)

Update-CMAMTProvisioning

Update-CMAMTProvisioning

Updates provisioning for an Intel AMT-based computer.

Syntax

Parameter Set: SearchByNameMandatory

```
Update-CMAMTProvisioning -DeviceName <String[]> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByIdMandatory

```
Update-CMAMTProvisioning -DeviceId <String[]> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Update-CMAMTProvisioning -Device <IResultObject> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Update-CMAMTProvisioning** cmdlet updates provisioning for an Intel Active Management Technology (Intel AMT)-based computer. Provisioning is the process of initializing and registering a computer with Intel AMT technology with Microsoft System Center 2012 Configuration Manager for out-of-band management. This cmdlet updates provisioning information.

You can specify computers to update by using the System Center 2012 Configuration Manager device name or device ID, or you can use the **Get-CMDevice** cmdlet to get a device object.

Parameters

-Device<IResultObject>

Specifies a device object. To obtain a device object, use the **Get-CMDevice** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-DeviceId<String[]>

Specifies an array of IDs of devices.

Aliases	ResourceId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeviceName<String[]>

Specifies an array of names of devices.

Aliases	Name
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Update provisioning for a device by using an ID

This command updates provisioning for an Intel AMT-based computer that has the device ID 16777230.

```
PS C:\> Update-CMAmtProvisioning -DeviceID "16777230"
```

Example 2: Update provisioning for a device by using an ID

This command updates provisioning for an Intel AMT-based computer named Accn023.Contoso.com.

```
PS C:\> Update-CMAmtProvisioning -DeviceName "Accn023.Contoso.com"
```

Example 3: Enable audit logging by using a variable

The first command gets a device object by using the **Get-CMDevice** command, and stores it in the \$CMD variable.

The second command updates provisioning for the device stored in \$CMD.

```
PS C:\> $CMD = Get-CMDevice -Name "Accn023.Contoso.com"  
PS C:\> Update-CMAmtProvisioning -Device $CMD
```

Related topics

[Invoke-CMAmtProvisioningDiscovery](#)

[Remove-CMAmtProvisioningData](#)

[Get-CMDevice](#)

Update-CMApplicationStatistic

Update-CMApplicationStatistic

Updates the statistics for an application.

Syntax

Parameter Set: SearchByIdMandatory

```
Update-CMApplicationStatistic -Id <String[]> [ <CommonParameters>]
```

Parameter Set: SearchByNameMandatory

```
Update-CMApplicationStatistic -Name <String[]> [ <CommonParameters>]
```

Parameter Set: SearchByValueMandatory

```
Update-CMApplicationStatistic -InputObject <IResultObject> [ <CommonParameters>]
```

Detailed Description

The **Update-CMApplicationStatistic** cmdlet updates the statistics for a Microsoft System Center 2012 Configuration Manager application.

Parameters

-Id<String[]>

Specifies an array of IDs of applications.

Aliases	CId
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-InputObject<IResultObject>

Specifies a Configuration Manager application statistic object.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Name<String[]>

Specifies an array of names of applications.

Aliases	LocalizedDisplayName
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

<CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

Examples

Example 1: Update statistics for an application by ID

This command updates statistics for an application that has the ID 16781415.

```
PS C:\> Update-CMAApplicationStatistic -Id "16781415"
```

Example 2: Update statistics for an application by name

This command updates statistics for an application named Test.

```
PS C:\> Update-CMApplicationStatistic -Name "Test"
```

Example 3: Update statistics for an application by name by using a variable

The first command gets the application object named Test and stores the object in the \$App variable.

The second command updates statistics for the application stored in \$App.

```
PS C:\> $App = Get-CMApplication -Name "Test"
```

```
PS C:\> Update-CMApplicationStatistic -InputObject $App
```

Related topics

[Get-CMApplication](#)

Update-CMCertificate

Update-CMCertificate

Updates a certificate.

Syntax

Parameter Set: Default

```
Update-CMCertificate -Id <String> -Path <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Update-CMCertificate** cmdlet updates a public key infrastructure (PKI) certificate that Microsoft System Center 2012 Configuration Manager uses.

Parameters

-Id<String>

Specifies the ID of a certificate.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Path<String>

Specifies a certification path.

Aliases	none
---------	------

Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Update a certificate

This command updates the certificate that has the ID BaseCert.

```
PS C:\> Update-CMCertificate -Id "BaseCert.txt" -Path "\\Contoso01\CM\Toolbox\BaseCert.txt"
```

Related topics

[Block-CMCertificate](#)

[Unblock-CMCertificate](#)

[Import-CMCertificate](#)

Update-CMClientStatus

Update-CMClientStatus

Updates client status.

Syntax

Parameter Set: Default

```
Update-CMClientStatus [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Update-CMClientStatus** cmdlet updates the client status on the local computer.

Parameters

-Force

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Update client status settings

This command updates the client status on the local computer.

```
PS C:\> Update-CMClientStatus
```

Related topics

[Get-CMClientStatusSetting](#)

[Set-CMClientStatusSetting](#)

[About Client Settings in Configuration Manager](#)

Update-CMDistributionPoint

Update-CMDistributionPoint

Updates distribution points with the latest content.

Syntax

Parameter Set: SearchByBootImage

```
Update-CMDistributionPoint -BootImage <IResultObject> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByBootImageId

```
Update-CMDistributionPoint -BootImageId <String> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SearchByBootImageName

```
Update-CMDistributionPoint -BootImageName <String> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByDriverPackage

```
Update-CMDistributionPoint -DriverPackage <IResultObject> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByDriverPackageId

```
Update-CMDistributionPoint -DriverPackageId <String> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByDriverPackageName

```
Update-CMDistributionPoint -DriverPackageName <String> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByOSImage

```
Update-CMDistributionPoint -OperatingSystemImage <IResultObject> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByOSImageId

```
Update-CMDistributionPoint -OperatingSystemImageId <String> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByOSImageName

```
Update-CMDistributionPoint -OperatingSystemImageName <String> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByOSInstaller

```
Update-CMDistributionPoint -OperatingSystemInstaller <IResultObject> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: SearchByOSInstallerId

Update-CMDistributionPoint -OperatingSystemInstallerId <String> [-Confirm] [-WhatIf] [
<CommonParameters>]

Parameter Set: SearchByOSInstallerName

Update-CMDistributionPoint -OperatingSystemInstallerName <String> [-Confirm] [-WhatIf] [
<CommonParameters>]

Parameter Set: SearchByPackage

Update-CMDistributionPoint -Package <IResultObject> [-Confirm] [-WhatIf] [
<CommonParameters>]

Parameter Set: SearchByPackageId

Update-CMDistributionPoint -PackageId <String> [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SearchByPackageName

Update-CMDistributionPoint -PackageName <String> [-Confirm] [-WhatIf] [<CommonParameters>]

Parameter Set: SearchBySoftwareUpdateDeploymentPackage

Update-CMDistributionPoint -SoftwareUpdateDeploymentPackage <IResultObject> [-Confirm] [-
WhatIf] [<CommonParameters>]

Parameter Set: SearchBySoftwareUpdateDeploymentPackageId

Update-CMDistributionPoint -SoftwareUpdateDeploymentPackageId <String> [-Confirm] [-WhatIf]
[<CommonParameters>]

Parameter Set: SearchBySoftwareUpdateDeploymentPackageName

Update-CMDistributionPoint -SoftwareUpdateDeploymentPackageName <String> [-Confirm] [-
WhatIf] [<CommonParameters>]

Parameter Set: UpdateByDeploymentTypeName

Update-CMDistributionPoint -ApplicationName <String> -DeploymentTypeName <String> [-
ManifestPath <String>] [-Confirm] [-WhatIf] [<CommonParameters>]

Detailed Description

The **Update-CMDistributionPoint** cmdlet updates distribution points with the latest content for clients to download. You can update the distribution points for application content, software packages, software updates, operating system images, and boot images. Manually updating the distribution points does not interfere with the recurring update schedule.

Parameters

-ApplicationName<String>

Specifies the name of an application that is associated with the deployment type.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-BootImage<IResultObject>

Specifies a **CMBootImage** object. To obtain a **CMBootImage** object, use the **Get-CMBootImage** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-BootImageId<String>

Specifies the ID of a boot image.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-BootImageName<String>

Specifies the name of a boot image.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DeploymentTypeName<String>

Specifies the name of a deployment type. A deployment type is contained within an application and contains the information that Configuration Manager requires to install software. A deployment type also contains rules that specify if and how Configuration Manager deploys the software.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DriverPackage<IResultObject>

Specifies a **CMDriverPackage** object. To obtain a **CMDriverPackage** object, use the **Get-CMDriverPackage** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DriverPackageId<String>

Specifies the ID of a driver package.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-DriverPackageName<String>

Specifies the name of a driver package.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-ManifestPath<String>

Specifies the Universal Naming Convention (UNC) path of the virtual application's XML manifest file. Specify the manifest path if you specify Microsoft Application Virtualization for the *DeploymentTypeName* parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OperatingSystemImage<IResultObject>

Specifies a **CMOperatingSystemImage** object. To obtain a **CMOperatingSystemImage** object, use the **Get-CMOperatingSystemImage** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OperatingSystemImageId<String>

Specifies the ID of an operating system image.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OperatingSystemImageName<String>

Specifies the name of an operating system image.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

-OperatingSystemInstaller<IResultObject>

Specifies a **CMOperatingSystemInstaller** object. To obtain a **CMOperatingSystemInstaller** object, use the **Get-CMOperatingSystemInstaller** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OperatingSystemInstallerId<String>

Specifies the ID of an operating system installer.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-OperatingSystemInstallerName<String>

Specifies the name of an operating system installer.

Aliases	none
Required?	true
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-Package<IResultObject>

Specifies a **CMPackage** object. To obtain a **CMPackage** object, use the **Get-CMPackage** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PackageId<String>

Specifies the ID of a package.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-PackageName<String>

Specifies the name of a package.

Aliases	none
Required?	true
Position?	named
Default Value	none

Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SoftwareUpdateDeploymentPackage<IResultObject>

Specifies a **CMSoftwareUpdateDeploymentPackage** object. To obtain a **CMSoftwareUpdateDeploymentPackage** object, use the **Get-CMSoftwareUpdateDeploymentPackage** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SoftwareUpdateDeploymentPackageId<String>

Specifies the ID of a software update deployment.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

-SoftwareUpdateDeploymentPackageName<String>

Specifies the name of a software update deployment.

Aliases	none
Required?	true

Position?	named
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
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

Example 1: Update the distribution points for a deployment type

This command updates the distribution points for the deployment type named 7Zip - Microsoft Application Virtualization 5. The command specifies that the Microsoft Application Virtualization manifest for the deployment type is contained in 7Zip.appv.

```
PS C:\> Update-CMDistributionPoint -DeploymentTypeName "7Zip - Microsoft Application Virtualization 5" -ManifestPath "\\Contoso01\DeployType\7Zip\7Zip.appv"
```

Related topics

[Get-CMDistributionPoint](#)

[Set-CMDistributionPoint](#)

[Remove-CMDistributionPoint](#)

[Get-CMDistributionPointGroup](#)

[Get-CMBootImage](#)

[Get-CMDriverPackage](#)

[Get-CMOperatingSystemImage](#)

[Get-CMOperatingSystemInstaller](#)

[Get-CMPackage](#)

[Get-CMSoftwareUpdateDeploymentPackage](#)