



# System Center 2012 – Operations Manager Cmdlet Reference

---

Microsoft Corporation

Published: April 2012

## Applies To

System Center 2012 – Operations Manager

## About This Document

This downloadable document contains the help topics for the Operations Manager cmdlets. For the most current documentation about System Center 2012 – Operations Manager cmdlets, see [Cmdlets in System Center 2012 - Operations Manager](#) in the TechNet Library.

## Feedback

Send suggestions and comments about Operations Manager documentation to [opsmgrdocs@microsoft.com](mailto:opsmgrdocs@microsoft.com).

# Copyright

---

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

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

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

© 2012 Microsoft Corporation. All rights reserved.

Microsoft, Active Directory, Bing, Internet Explorer, JScript, SharePoint, Silverlight, SQL Server, Visio, Visual Basic, Visual Studio, Win32, Windows, Windows Intune, Windows PowerShell, and Windows Vista are trademarks of the Microsoft group of companies. Portions of this documentation related to network monitoring are provided by EMC, and for those portions the following copyright notice applies 2010 © EMC Corporation. All rights reserved. All other trademarks are property of their respective owners.

# Revision History

---

Release Date	Changes
April 1, 2012	Original release of this document.

## Contents

Add-SCOMADAgentAssignment .....	9
Add-SCOMAgentlessManagedComputer.....	14
Add-SCOMAlertResolutionState .....	16
Add-SCOMConnector .....	20
Add-SCOMNotificationChannel .....	24
Add-SCOMNotificationSubscriber .....	37
Add-SCOMNotificationSubscription.....	41
Add-SCOMRunAsAccount.....	49
Add-SCOMRunAsProfile .....	63
Add-SCOMSubscriberSchedule .....	68
Add-SCOMTierConnector.....	74
Add-SCOMTieredManagementGroup .....	78
Add-SCOMUserRole .....	84
Approve-SCOMPendingManagement .....	91
Clear-SCOMSubscriberSchedule .....	94
Deny-SCOMPendingManagement .....	98
Disable-SCOMAgentProxy .....	101
Disable-SCOMDiscovery .....	104
Disable-SCOMMonitor .....	108
Disable-SCOMNotificationSubscription .....	112
Disable-SCOMOperationalDataReporting.....	115
Disable-SCOMRule .....	117
Enable-SCOMAgentProxy .....	122
Enable-SCOMDiscovery.....	125
Enable-SCOMMonitor.....	130

Enable-SCOMNotificationSubscription .....	135
Enable-SCOMOperationalDataReporting.....	137
Enable-SCOMRule .....	139
Exit-SCOMCEIP .....	144
Export-SCOMEffectiveMonitoringConfiguration .....	147
Export-SCOMManagementPack .....	151
Get-SCOMAccessLicense .....	154
Get-SCOMADAgentAssignment.....	158
Get-SCOMAgent.....	161
Get-SCOMAgentApprovalSetting .....	165
Get-SCOMAgentlessManagedComputer .....	167
Get-SCOMAlert.....	171
Get-SCOMAlertHistory .....	178
Get-SCOMAlertResolutionSetting .....	180
Get-SCOMAlertResolutionState .....	182
Get-SCOMClass .....	185
Get-SCOMClassInstance .....	190
Get-SCOMCommand .....	195
Get-SCOMConnector .....	198
Get-SCOMDatabaseGroomingSetting .....	202
Get-SCOMDataWarehouseSetting.....	204
Get-SCOMDiagnostic .....	206
Get-SCOMDiscovery .....	211
Get-SCOMErrorReportingSetting .....	216
Get-SCOMEvent.....	218
Get-SCOMGatewayManagementServer .....	223
Get-SCOMGroup .....	226

Get-SCOMHeartbeatSetting .....	230
Get-SCOMLicense.....	232
Get-SCOMLocation .....	234
Get-SCOMMaintenanceMode .....	240
Get-SCOMManagementGroup .....	244
Get-SCOMManagementGroupConnection.....	247
Get-SCOMManagementPack .....	250
Get-SCOMManagementServer .....	255
Get-SCOMMonitor .....	258
Get-SCOMNotificationChannel.....	265
Get-SCOMNotificationSubscriber .....	268
Get-SCOMNotificationSubscription .....	271
Get-SCOMOverride .....	275
Get-SCOMOverrideResult.....	282
Get-SCOMPARENTManagementServer.....	288
Get-SCOMPendingManagement.....	290
Get-SCOMRecovery.....	293
Get-SCOMRelationship .....	298
Get-SCOMRelationshipInstance .....	304
Get-SCOMReportingSetting .....	308
Get-SCOMResourcePool .....	310
Get-SCOMRMSEmulator.....	316
Get-SCOMRule.....	319
Get-SCOMRunAsAccount .....	325
Get-SCOMRunAsDistribution .....	328
Get-SCOMRunAsProfile .....	331
Get-SCOMTask .....	336

Get-SCOMTaskResult .....	342
Get-SCOMTierConnector .....	347
Get-SCOMTieredManagementGroup .....	351
Get-SCOMUserRole .....	355
Get-SCOMWebAddressSetting .....	359
Import-SCOMManagementPack .....	361
Install-SCOMAgent .....	366
Join-SCOMCEIP .....	371
New-SCOMLocation .....	374
New-SCOMManagementGroupConnection .....	378
New-SCOMResourcePool .....	381
Remove-SCOMADAgentAssignment .....	385
Remove-SCOMAgentlessManagedComputer .....	389
Remove-SCOMAlertResolutionState .....	391
Remove-SCOMConnector .....	394
Remove-SCOMDisabledClassInstance .....	397
Remove-SCOMLocation .....	400
Remove-SCOMManagementGroupConnection .....	406
Remove-SCOMManagementPack .....	409
Remove-SCOMNotificationChannel .....	412
Remove-SCOMNotificationSubscriber .....	416
Remove-SCOMNotificationSubscription .....	419
Remove-SCOMResourcePool .....	422
Remove-SCOMRMSEmulator .....	426
Remove-SCOMRunAsAccount .....	429
Remove-SCOMRunAsProfile .....	432
Remove-SCOMSubscriberSchedule .....	435

Remove-SCOMTierConnector.....	439
Remove-SCOMTieredManagementGroup .....	443
Repair-SCOMAgent.....	446
Resolve-SCOMAlert .....	448
Set-SCOMAgentApprovalSetting .....	455
Set-SCOMAgentlessManagedComputer.....	459
Set-SCOMAlert .....	462
Set-SCOMAlertResolutionSetting.....	469
Set-SCOMDatabaseGroomingSetting .....	473
Set-SCOMDataWarehouseSetting .....	479
Set-SCOMErrorReportingSetting .....	483
Set-SCOMHeartbeatSetting .....	488
Set-SCOMLicense .....	492
Set-SCOMLocation.....	495
Set-SCOMMaintenanceMode.....	500
Set-SCOMManagementGroupConnection .....	504
Set-SCOMParentManagementServer .....	506
Set-SCOMReportingSetting.....	511
Set-SCOMResourcePool.....	515
Set-SCOMRMSEmulator .....	522
Set-SCOMRunAsDistribution.....	525
Set-SCOMRunAsProfile .....	531
Set-SCOMUserRole .....	536
Set-SCOMWebAddressSetting.....	542
Start-SCOMMaintenanceMode.....	546
Start-SCOMTask .....	549
Test-SCOMCEIP .....	553

Test-SCOMOperationalDataReporting .....	555
Uninstall-SCOMAgent.....	558
Update-SCOMADAgentAssignment.....	561
Update-SCOMLocation.....	566
Update-SCOMRunAsAccount .....	570
Write-SCOMCommand.....	578



# Add-SCOMADAgentAssignment

---

## Add-SCOMADAgentAssignment

Adds a new Active Directory agent assignment to the management group.

### Syntax

Parameter Set: Default

```
Add-SCOMADAgentAssignment [-Domain] <String> [-PrimaryServer] <ManagementServer> [-LdapQuery] <String> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-Exclude <String[]> ] [-FailoverServer <ManagementServer[]> ] [-RunAsProfile <ManagementPackSecureReference> ] [-SCSession <Connection[]> ] [-Confirm] [-WhatIf] [ <CommonParameters> ]
```

### Detailed Description

The Add-SCOMADAgentAssignment cmdlet adds a new Active Directory agent assignment to the management group.

Before running this command, prepare the agent AD environment by executing the MOMADAdmin.exe utility from a computer in the agent domain.

### Parameters

#### -ComputerName<String[]>

Specifies a computer to establish a connection with. The computer must be running the System Center Data Access service. The default is the computer for the current management group connection.

Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

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

## **-Credential<PSCredential>**

Specifies a user account under which the management group connection will run. It must have access to the server that is specified in the ComputerName parameter if specified. The default is the current user.

Enter a PSCredential object, such as one that is returned by the Get-Credential cmdlet.

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

## **-Domain<String>**

The domain name or domain controller name for the domain in which the target agents reside.

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

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

A list of computer names which should be excluded from the agent assignment, even if they are returned by the LDAP query.

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

## **-FailoverServer<ManagementServer[]>**

A list of 1 or more management servers used as failovers for the target agents

By default all non-gateway management servers will be used for failover

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

## **-LdapQuery<String>**

The LDAP query in the domain which specifies the desired target agent computers.

Aliases	none
Required?	true
Position?	3
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-PrimaryServer<ManagementServer>**

The management server which the target agents will report to.

Aliases	none
Required?	true
Position?	2
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-RunAsProfile<ManagementPackSecureReference>**

The runas profile used when executing the AD query workflow.

The accounts specified in this profile should be the same as those specified when executing MOMADAdmin.exe.

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

## **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection.

Enter a Management group connection object, such as one returned by the Get-SCManagementGroupConnection cmdlet.

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

## **-Confirm**

Prompts you for confirmation before executing the command.

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

## -WhatIf

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

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

## <CommonParameters>

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

## Examples

### ----- EXAMPLE 1 -----

This example adds a new AD agent assignment in the fabrikam.net domain for computers with names like SQLServer\*.fabrikam.net but excludes SQLServer05.fabrikam.net

```
PS C:\>$primaryServer = Get-SCOMManagementServer "FabrikamMS.fabrikam.net"
PS C:\>$ldap = '(&(sAMAccountType=805306369)(name=SQLServer*))'
PS C:\>Add-SCOMADAgentAssignment -Domain 'fabrikam.net' -PrimaryServer $primaryServer -
LdapQuery $ldap -Exclude 'SQLServer05.fabrikam.net'
```

### ----- EXAMPLE 2 -----

This example adds a new AD agent assignment in the fabrikam.net domain for computers with names like SQLServer\*.fabrikam.net and includes failover servers and uses a different Run as profile.

```
PS C:\>$primaryServer = Get-SCOMManagementServer "FabrikamMS.fabrikam.net"
PS C:\>$failovers = Get-SCOMManagementServer "FabrikamSecondary*.fabrikam.net"
PS C:\>$runAs = Get-SCOMRunAsProfile "Active Directory Based Agent Assignment Account"
PS C:\>$ldap = '(&(sAMAccountType=805306369)(name=SQLServer*))'
PS C:\>Add-SCOMADAgentAssignment -Domain 'fabrikam.net' -PrimaryServer $primaryServer -
LdapQuery $ldap -FailoverServer $failovers -RunAsProfile $runas
```

# Add-SCOMAgentlessManagedComputer

---

## Add-SCOMAgentlessManagedComputer

Adds managed computers that do not have agents to a management group.

### Syntax

Parameter Set: FromAgentManagedBy

```
Add-SCOMAgentlessManagedComputer [-DNSHostName] <String[]> [-ManagedByAgent]  
<AgentManagedComputer> [ <CommonParameters>]
```

Parameter Set: FromManagementServerManagedBy

```
Add-SCOMAgentlessManagedComputer [-DNSHostName] <String[]> [-ManagedByManagementServer]  
<ManagementServer> [ <CommonParameters>]
```

### Detailed Description

The Add-SCOMAgentlessManagedComputer cmdlet adds managed computers that do not have agents to a management group.

### Parameters

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

Specifies the name of a Domain Name System (DNS) host.

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

#### **-ManagedByAgent<AgentManagedComputer>**

Specifies the agent which will perform the agentless management.

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

Required?	true
Position?	2
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-ManagedByManagementServer<ManagementServer>**

Specifies the management server that will perform the agentless management.

Aliases	none
Required?	true
Position?	2
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

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

## **Examples**

### **----- EXAMPLE 1 -----**

This command specifies that management server MgmtServer01 will perform agentless management for Server01. You will be prompted to confirm the operation before the command is run.

```
PS C:\>Add-SCOMAgentlessManagedComputer -Name Server01.Contoso.com -
ManagedByManagementServer (Get-SCOMManagementServer -Name MgmtServer01.Contoso.com) -Confirm
```

# Add-SCOMAlertResolutionState

---

## Add-SCOMAlertResolutionState

Adds a custom alert resolution state.

### Syntax

Parameter Set: FromManagementState

```
Add-SCOMAlertResolutionState -Name <String> -ResolutionStateCode <Byte> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [-ShortcutKey <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The Add-SCOMAlertResolutionState cmdlet adds a custom alert resolution state.

### Parameters

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

Specifies a computer to establish a connection with. The computer must be running the System Center Data Access service. The default is the computer for the current management group connection.

Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

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

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

Specifies a user account under which the management group connection will run. It must have access to the server that is specified in the ComputerName parameter if specified. The default is the current user.

Enter a PSCredential object, such as one that is returned by the Get-Credential cmdlet.



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

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

The friendly name of the alert resolution state.

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

### **-ResolutionStateCode<Byte>**

The resolution state code. Two states are defined by the system: New (0) and Closed (255). Any other value can be used.

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

### **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection.

Enter a Management group connection object, such as one returned by the Get-SCManagementGroupConnection cmdlet.

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

### **-ShortcutKey<String>**

A shortcut key used for the resolution state.

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 -----

This example adds a new alert resolution state called Investigating, with code 10

```
PS C:\>Add-SCOMAlertResolutionState -Name Investigating -ResolutionStateCode 10
```

# Add-SCOMConnector

---

## Add-SCOMConnector

Adds a connector.

### Syntax

Parameter Set: Empty

```
Add-SCOMConnector [-Name] <String> [[-DisplayName] <String> ] [[-Description] <String> ] [[-Guid] <Guid> ] [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-DiscoveryDataIsManaged] [-DiscoveryDataIsShared] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

### Detailed Description

The Add-SCOMConnector cmdlet adds a connector. Operations Manager connectors are used to connect Operations Manager to other products.

### Parameters

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

Specifies a computer to establish a connection with. The computer must be running the System Center Data Access service. The default is the computer for the current management group connection.

Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

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

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

Specifies a user account under which the management group connection will run. It must have access to the server that is specified in the ComputerName parameter if specified. The default is the current user.

Enter a PSCredential object, such as one that is returned by the Get-Credential cmdlet.

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

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

The description of the connector. Will default to the displayname.

Aliases	none
Required?	false
Position?	3
Default Value	none
Accept Pipeline Input?	true (ByPropertyName)
Accept Wildcard Characters?	false

### **-DiscoveryDatalsManaged**

If set, discovery data will be managed.

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

### **-DiscoveryDatalsShared**

If set, discovery data will be shared.

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

### **-DisplayName<String>**

The display name of the connector. Will default to the name.

Aliases	none
Required?	false
Position?	2
Default Value	none
Accept Pipeline Input?	true (ByPropertyName)
Accept Wildcard Characters?	false

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

The GUID of the new connector.

Aliases	none
Required?	false
Position?	4
Default Value	none
Accept Pipeline Input?	true (ByPropertyName)
Accept Wildcard Characters?	false

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

The name of the connector

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

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

### **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection.

Enter a Management group connection object, such as one returned by the Get-SCManagementGroupConnection cmdlet.

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

### **<CommonParameters>**

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

### **Examples**

#### **----- EXAMPLE 1 -----**

This example adds a new connector.

```
PS C:\>Add-SCOMConnector -Name MyCustomConnector -DisplayName "Connector for shared data" -
Description "Used to forward data to other programs"
```

# Add-SCOMNotificationChannel

---

## Add-SCOMNotificationChannel

Adds a notification channel.

### Syntax

Parameter Set: UsingSMTP

```
Add-SCOMNotificationChannel [-Name] <String> [-From] <String> -Body <String> -Server <String> [-Anonymous] [-BackupSmtpServer <String[]> ] [-BodyAsHtml] [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-Description <String> ] [-DisplayName <String> ] [-Encoding <String> ] [-Header <Hashtable> ] [-MaxPrimaryRecipientsPerMail <Int32> ] [-NoSubjectEncoding] [-Port <UInt32> ] [-PrimaryRetryInterval <TimeSpan> ] [-ReplyTo <String> ] [-SCSession <Connection[]> ] [-Subject <String> ] [ <CommonParameters>]
```

Parameter Set: CommandAction

```
Add-SCOMNotificationChannel [-ApplicationPath] <String> [-Name] <String> [[-Argument] <String> ] [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-Description <String> ] [-DisplayName <String> ] [-SCSession <Connection[]> ] [-WorkingDirectory <String> ] [ <CommonParameters>]
```

Parameter Set: UsingSIP

```
Add-SCOMNotificationChannel [-Name] <String> [-UserName] <Uri> -Body <String> -Server <String> [-ComputerName <String[]> ] [-ContentType <String> ] [-Credential <PSCredential> ] [-Description <String> ] [-DisplayName <String> ] [-Encoding <String> ] [-Port <UInt32> ] [-PreferredUserName <String> ] [-SCSession <Connection[]> ] [-SipAuthentication <SipNotificationAuthenticationProtocols> ] [-SipProtocol <SipTransportProtocol> ] [ <CommonParameters>]
```

Parameter Set: UsingSMS

```
Add-SCOMNotificationChannel [-Name] <String> [[-Device] <String> ] [[-BackupDevice] <String[]> ] -Body <String> -Sms[-ComputerName <String[]> ] [-Credential <PSCredential> ] [-Description <String> ] [-DeviceEnumerationInterval <TimeSpan> ] [-DisplayName <String> ] [-Encoding <String> ] [-PrimaryRetryInterval <TimeSpan> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

### Detailed Description

The Add-SCOMNotification cmdlet adds a notification channel.

Notification channels inform individuals when an alert happens, or run automation in response to an alert.

### Parameters

#### -Anonymous

If set, will use an anonymous SMTP server.



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

### **-ApplicationPath<String>**

The application path of a command channel.

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

### **-Argument<String>**

The string of arguments to a command channel.

Aliases	none
Required?	false
Position?	2
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

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

Secondary SMS devices.

Aliases	none
Required?	false
Position?	3
Default Value	none
Accept Pipeline Input?	true (ByPropertyName)
Accept Wildcard Characters?	false

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

One or more backup SMTP servers.

Backup servers will use the same configuration as the primary server.

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

### **-Body<String>**

The body or content of the notification.

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

### **-BodyAsHtml**

If set, the notification will be an HTML mail.

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

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

Specifies a computer to establish a connection with. The computer must be running the System Center Data Access service. The default is the computer for the current management group connection.

Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

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

### **-ContentType<String>**

The content type of the instant message. The default is 'text/plain'

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

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

Specifies a user account under which the management group connection will run. It must have access to the server that is specified in the ComputerName parameter if specified. The default is the current user.

Enter a PSCredential object, such as one that is returned by the Get-Credential cmdlet.

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

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

A description for the notification channel. If not set, defaults to the displayname.

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

### **-Device<String>**

The primary SMS device.

Aliases	none
Required?	false
Position?	2
Default Value	none
Accept Pipeline Input?	true (ByPropertyName)
Accept Wildcard Characters?	false

## **-DeviceEnumerationInterval<TimeSpan>**

The device enumeration interval, for SMS notifications.

The default is 10 seconds

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

## **-DisplayName<String>**

The display name for the notification channel. If not set, defaults to the name.

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

## **-Encoding<String>**

The encoding to use for the notification.

SMTP channels default to 'utf-8' encoding.

IM channels default to 'utf-8' encoding.

SMS channels use 'Default' encoding by default, but can alternately use 'Unicode' encoding.

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

## **-From<String>**

The from field in a notification email.

Aliases	none
Required?	true
Position?	2
Default Value	none
Accept Pipeline Input?	true (ByPropertyName)
Accept Wildcard Characters?	false

## **-Header<Hashtable>**

Any headers in a notification email.

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

## **-MaxPrimaryRecipientsPerMail<Int32>**

The maximum number of recipients per notification mail.

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

## **-Name<String>**

The name of the notification channel.

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

## **-NoSubjectEncoding**

If set, will not use an encoding for the email subject.

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

## **-Port<UInt32>**

The port used for the channel.

SMTP channels default to port 25

IM channels using TCP default to port 5060

IM channels using TLS default to port 5061

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

## **-PreferredUserName<String>**

The preferred username in an IM channel. If not set, defaults to the username.

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

## **-PrimaryRetryInterval<TimeSpan>**

The retry interval used to attempt to switch back to the primary SMTP server or SMS device. The default is 5 minutes.

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

## **-ReplyTo<String>**

The reply-to field in a notification email.

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



## **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection.

Enter a Management group connection object, such as one returned by the Get-SCManagementGroupConnection cmdlet.

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

## **-Server<String>**

The server used for SMTP and IM channels

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

## **-SipAuthentication<SipNotificationAuthenticationProtocols>**

The SIP authentication mechanism (NTLM or Kerberos). Used for IM channels. The default is NTLM.

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

## **-SipProtocol<SipTransportProtocol>**

The SIP transport protocol (TCP or TLS). Used for IM channels. The default is TCP.

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

## **-Sms**

If set, will send notifications using SMS

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

## **-Subject<String>**

The subject of the notification mail.

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

## **-UserName<Uri>**

The IM return address

Aliases	none
Required?	true
Position?	2
Default Value	none
Accept Pipeline Input?	true (ByPropertyName)
Accept Wildcard Characters?	false

## **-WorkingDirectory<String>**

The working directory used for a command channel. Defaults to "%systemdrive%"

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 -----**

This example adds a standard SMTP email channel.

```
PS C:\>$subject = "SCOM alert `"$Data[Default='Not Present']/Context/DataItem/AlertName`"
```

```
PS C:\>$body = "Owner is `"$Data[Default='Not Present']/Context/DataItem/AlertOwner`"
```

```
PS C:\>Add-SCOMNotificationChannel -Name "Contoso.Email" -Server "mail.contoso.com" -From "scom@contoso.net" -Subject $subject -Body $body
```

## ----- EXAMPLE 2 -----

This example adds a new IM channel, with the return address of "scomadmin".

```
PS C:\>$body = "SCOM alert `"$Data[Default='Not Present']/Context/DataItem/AlertName`$"
PS C:\>Add-SCOMNotificationChannel -Name "Contoso.IM" -Server "sipserver.contoso.com" -
UserName 'sip:scomadmin' -Body $body
```

## ----- EXAMPLE 3 -----

This example adds a new SMS channel with unicode encoding.

```
PS C:\>$body = "SCOM alert `"$Data[Default='Not Present']/Context/DataItem/AlertName`$"
PS C:\>Add-SCOMNotificationChannel -Sms -Name "Contoso.SMS" -Body $body -Encoding "Unicode"
```

## ----- EXAMPLE 4 -----

This example adds a new command notification channel which runs the OnNotify.exe program.

```
PS C:\> $path = "C:\OnNotify.exe"
PS C:\>$arg = "/notify /owner `"$Data[Default='Not
Present']/Context/DataItem/AlertOwner``"
PS C:\>$workingDir = "C:\\"
PS C:\>Add-SCOMNotificationChannel -Name "Contoso.Command" -ApplicationPath $path -Argument
$arg -WorkingDirectory $workingDir
```

# Add-SCOMNotificationSubscriber

---

## Add-SCOMNotificationSubscriber

Adds a SCOM notification subscriber.

### Syntax

Parameter Set: DeviceName

```
Add-SCOMNotificationSubscriber [-Name] <String> [-DeviceList] <String[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: DeviceTable

```
Add-SCOMNotificationSubscriber [-Name] <String> [-DeviceTable] <Hashtable> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

### Detailed Description

The Add-SCOMNotificationSubscriber adds an Operations Manager notification subscriber.

Notification subscribers can receive notifications from SCOM.

### Parameters

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

Specifies a computer to establish a connection with. The computer must be running the System Center Data Access service. The default is the computer for the current management group connection.

Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

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

## **-Credential<PSCredential>**

Specifies a user account under which the management group connection will run. It must have access to the server that is specified in the ComputerName parameter if specified. The default is the current user.

Enter a PSCredential object, such as one that is returned by the Get-Credential cmdlet.

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

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

The list of notification addresses for the subscriber.

Email addresses should be of the form 'username@domain.com'

SMS addresses should be of the form 'SMS:<address>'

IM addresses should be of the form 'SIP:<address>'

Command channels can be specified by specifying the name of the channel.

Aliases	none
Required?	true
Position?	2
Default Value	none
Accept Pipeline Input?	true (ByPropertyName)
Accept Wildcard Characters?	false

## **-DeviceTable<Hashtable>**

A table of named addresses.

Email addresses should be of the form 'username@domain.com'

SMS addresses should be of the form 'SMS:<address>'

IM addresses should be of the form 'SIP:<address>'

Command channels can be specified by specifying the name of the channel.

Aliases	none
Required?	true
Position?	2
Default Value	none
Accept Pipeline Input?	true (ByPropertyName)
Accept Wildcard Characters?	false

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

The name of the notification subscriber.

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

### **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection.

Enter a Management group connection object, such as one returned by the Get-SCManagementGroupConnection cmdlet.

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

## <CommonParameters>

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

## Examples

### ----- EXAMPLE 1 -----

This example adds a new notification subscriber with email, SMS, and IM addresses.

```
PS C:\>Add-SCOMNotificationSubscriber -Name 'John Doe' -DeviceList  
'john@contoso.com', 'sms:2065551212', 'sip:johnd'
```

### ----- EXAMPLE 2 -----

This example adds a new notification subscriber with an SMS address and a command address

```
PS C:\>$commandChannel = Get-SCOMNotificationChannel "OnNotify.exe"  
PS C:\>Add-SCOMNotificationSubscriber -Name 'Jane Doe' -DeviceTable @{ "Cell"=  
'sms:206555213'; "Command" = $commandChannel.Name }
```



# Add-SCOMNotificationSubscription

---

## Add-SCOMNotificationSubscription

Adds a new notification subscription.

### Syntax

Parameter Set: Default

```
Add-SCOMNotificationSubscription [-Name] <String> [[-Criteria] <String> ] -Channel  
<Object[]> -Subscriber <NotificationRecipient[]> [-BccSubscriber <NotificationRecipient[]> ]  
[-CcSubscriber <NotificationRecipient[]> ] [-ComputerName <String[]> ] [-Credential  
<PSCredential> ] [-Delay <TimeSpan> ] [-Description <String> ] [-Disabled] [-DisplayName  
<String> ] [-OnlyOnResolutionChange] [-PollingInterval <TimeSpan> ] [-SCSession  
<Connection[]> ] [ <CommonParameters>]
```

### Detailed Description

The Add-SCOMNotificationSubscription cmdlet adds a new notification subscription.

In this release there is no cmdlet support for scoping subscriptions based on a criteria. This can still be achieved via powershell by accessing the SCOM SDK directly.

See the examples for sample code which can aid with scoping subscriptions.

### Parameters

#### **-BccSubscriber<NotificationRecipient[]>**

BCCed subscribers for this subscription.

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

#### **-CcSubscriber<NotificationRecipient[]>**

CCed subscribers for this subscription.

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

### **-Channel<Object[]>**

The notification channel(s) to use. To obtain a notification channel object, use Get-SCOMNotificationChannel.

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

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

Specifies a computer to establish a connection with. The computer must be running the System Center Data Access service. The default is the computer for the current management group connection.

Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

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

## **-Credential<PSCredential>**

Specifies a user account under which the management group connection will run. It must have access to the server that is specified in the ComputerName parameter if specified. The default is the current user.

Enter a PSCredential object, such as one that is returned by the Get-Credential cmdlet.

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

## **-Criteria<String>**

The criteria xml indicating how to filter alerts for this subscription. Criteria can be used to filter on particular rules or monitors, or properties of the alert, but cannot be used to filter on classes or groups. In order to filter on classes or groups, see the examples.

The criteria xml can be manually specified or copied from an existing subscription. To obtain the criteria xml from an existing subscription, use this model: \$subscription = Get-SCOMNotificationSubscription | Select-Object -First 1 \$criteria = \$subscription.Configuration.Criteria

Aliases	none
Required?	false
Position?	3
Default Value	none
Accept Pipeline Input?	true (ByPropertyName)
Accept Wildcard Characters?	false

## **-Delay<TimeSpan>**

Delay sending notifications if conditions remain unchanged for this period of time.

By default, notifications will be sent immediately.

Aliases	none
Required?	false

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

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

The description of the subscription. Will use the displayname by default.

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

### **-Disabled**

If set, the subscription will be created in a disabled state.

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

### **-DisplayName<String>**

The display name of the subscription. Will use the name by default.

Aliases	none
Required?	false
Position?	named

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

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

The name of the subscription.

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

### **-OnlyOnResolutionChange**

If set, the notification will occur only when the resolution state of the alert changes.

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

### **-PollingInterval<TimeSpan>**

The polling interval for alerts. The default is 1 minute.

Aliases	none
Required?	false
Position?	named
Default Value	none

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

### **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection.

Enter a Management group connection object, such as one returned by the Get-SCManagementGroupConnection cmdlet.

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

### **-Subscriber<NotificationRecipient[]>**

The subscribers for this subscription. To obtain a notification subscriber object, use Get-SCOMNotificationSubscriber.

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 -----

This example adds a new notification subscription for all alerts.

```
PS C:\>$subscriber = Get-SCOMNotificationSubscriber "Neven Sokec","Renee Lo"
PS C:\>$channel = Get-SCOMNotificationChannel "EmailChannel"
PS C:\>Add-SCOMNotificationSubscription -Name "NewSubscription1" -Subscriber $subscriber -
Channel $channel
```

### ----- EXAMPLE 2 -----

This example adds a new notification subscription for all alerts, but only notifies if conditions remain unchanged for 1 hour. The example creates the subscription in the disabled state.

```
PS C:\>$subscriber = Get-SCOMNotificationSubscriber "Neven Sokec","Renee Lo"
PS C:\>$channel = Get-SCOMNotificationChannel "EmailChannel"
PS C:\>Add-SCOMNotificationSubscription -Name "NewSubscription2" -Subscriber $subscriber -
Channel $channel -Delay "1:00:00" -Disabled
```

### ----- EXAMPLE 3 -----

This example adds a new notification subscription for all alerts, then uses the Operations Manager SDK to scope the subscription to alerts raised by instances of the HealthService class.

```
PS C:\>$subscriber = Get-SCOMNotificationSubscriber "Neven Sokec","Renee Lo"
PS C:\>$channel = Get-SCOMNotificationChannel "EmailChannel"
PS C:\>$subscription = Add-SCOMNotificationSubscription -Name "NewSubscription3" -Subscriber
$subscriber -Channel $channel
PS C:\>$healthservice = Get-SCOMClass -Name "Microsoft.SystemCenter.HealthService"
PS C:\>$subscription.Configuration.MonitoringClassIds.Add( $healthservice.Id )
PS C:\>$subscription.Update()
```

### ----- EXAMPLE 4 -----

This example adds a new notification subscription for all alerts, then uses the Operations Manager SDK to scope the subscription to alerts raised by instances in the All Windows Computers group.

```
PS C:\>$subscriber = Get-SCOMNotificationSubscriber "Neven Sokec","Renee Lo"
PS C:\>$channel = Get-SCOMNotificationChannel "EmailChannel"
PS C:\>$subscription = Add-SCOMNotificationSubscription -Name "NewSubscription4" -Subscriber
$subscriber -Channel $channel
PS C:\>$windowsComputers = Get-SCOMGroup -DisplayName "All Windows Computers"
PS C:\>$subscription.Configuration.MonitoringObjectGroupIds.Add( $windowsComputers.Id )
PS C:\>$subscription.Update()
```

## ----- EXAMPLE 5 -----

This example adds a new notification subscription for all critical alerts raised by the "ContosoMonitor" monitor.

```
PS C:\>$subscriber = Get-SCOMNotificationSubscriber "Neven Sokec","Renee Lo"
PS C:\>$channel = Get-SCOMNotificationChannel "EmailChannel"
PS C:\>$monitor = Get-SCOMMonitor -Name "ContosoMonitor"
PS C:\>$criteria = @"
PS C:\><And xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
PS C:\><Expression>
PS C:\><SimpleExpression>
PS C:\><ValueExpression>
PS C:\><Property>ProblemId</Property>
PS C:\></ValueExpression>
PS C:\><Operator>Equal</Operator>
PS C:\><ValueExpression>
PS C:\><Value>$($monitor.Id)</Value>
PS C:\></ValueExpression>
PS C:\></SimpleExpression>
PS C:\></Expression>
PS C:\><Expression>
PS C:\><SimpleExpression>
PS C:\><ValueExpression>
PS C:\><Property>Severity</Property>
PS C:\></ValueExpression>
PS C:\><Operator>Equal</Operator>
PS C:\><ValueExpression>
PS C:\><Value>2</Value>
PS C:\></ValueExpression>
PS C:\></SimpleExpression>
PS C:\></Expression>
PS C:\></>
```



# Add-SCOMRunAsAccount

---

## Add-SCOMRunAsAccount

Adds a Run As account to the management group.

### Syntax

Parameter Set: Windows

```
Add-SCOMRunAsAccount [-Name] <String> [-RunAsCredential] <PSCredential> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-Description <String> ] [-SCSession <Connection[]> ] [-Windows] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ActionAccount

```
Add-SCOMRunAsAccount [-Name] <String> [-RunAsCredential] <PSCredential> [-ActionAccount] [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-Description <String> ] [-SCSession <Connection[]> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: Basic

```
Add-SCOMRunAsAccount [-Name] <String> [-RunAsCredential] <PSCredential> [-Basic] [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-Description <String> ] [-SCSession <Connection[]> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: Binary

```
Add-SCOMRunAsAccount [-Name] <String> [-Path] <String> [-Binary] [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-Description <String> ] [-SCSession <Connection[]> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: CommunityString

```
Add-SCOMRunAsAccount [-Name] <String> [-String] <SecureString> [-CommunityString] [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-Description <String> ] [-SCSession <Connection[]> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: Digest

```
Add-SCOMRunAsAccount [-Name] <String> [-RunAsCredential] <PSCredential> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-Description <String> ] [-Digest] [-SCSession <Connection[]> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SCXMaintenanceSSHKeyNoPrivSu

```
Add-SCOMRunAsAccount [-Name] <String> [-Path] <String> [-UserName] <String> -Su-SuPassword <SecureString> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-Description <String> ] [-Passphrase <SecureString> ] [-SCSession <Connection[]> ] [-SCXMaintenance] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SCXMaintenanceSSHKeyNoPrivSudo

```
Add-SCOMRunAsAccount [-Name] <String> [-Path] <String> [-UserName] <String> -Sudo[-ComputerName <String[]> ] [-Credential <PSCredential> ] [-Description <String> ] [-Passphrase <SecureString> ] [-SCSession <Connection[]> ] [-SCXMaintenance] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SCXMaintenanceSSHKeyPriv

```
Add-SCOMRunAsAccount [-Name] <String> [-Path] <String> [-UserName] <String> -Privileged[-ComputerName <String[]> ] [-Credential <PSCredential> ] [-Description <String> ] [-Passphrase <SecureString> ] [-SCSession <Connection[]> ] [-SCXMaintenance] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

```

Parameter Set: SCXMaintenanceUserPassNoPrivSu
Add-SCOMRunAsAccount [-Name] <String> [-RunAsCredential] <PSCredential> -Su-SuPassword
<SecureString> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-Description
<String> ] [-SCSession <Connection[]> ] [-SCXMaintenance] [-Confirm] [-WhatIf] [
<CommonParameters>]

Parameter Set: SCXMaintenanceUserPassNoPrivSudo
Add-SCOMRunAsAccount [-Name] <String> [-RunAsCredential] <PSCredential> -Sudo[-ComputerName
<String[]> ] [-Credential <PSCredential> ] [-Description <String> ] [-SCSession
<Connection[]> ] [-SCXMaintenance] [-Confirm] [-WhatIf] [ <CommonParameters>]

Parameter Set: SCXMaintenanceUserPassPriv
Add-SCOMRunAsAccount [-Name] <String> [-RunAsCredential] <PSCredential> -Privileged[-
ComputerName <String[]> ] [-Credential <PSCredential> ] [-Description <String> ] [-SCSession
<Connection[]> ] [-SCXMaintenance] [-Confirm] [-WhatIf] [ <CommonParameters>]

Parameter Set: SCXMonitoring
Add-SCOMRunAsAccount [-Name] <String> [-RunAsCredential] <PSCredential> [-ComputerName
<String[]> ] [-Credential <PSCredential> ] [-Description <String> ] [-SCSession
<Connection[]> ] [-SCXMonitoring] [-Sudo] [-Confirm] [-WhatIf] [ <CommonParameters>]

Parameter Set: Simple
Add-SCOMRunAsAccount [-Name] <String> [-RunAsCredential] <PSCredential> [-ComputerName
<String[]> ] [-Credential <PSCredential> ] [-Description <String> ] [-SCSession
<Connection[]> ] [-Simple] [-Confirm] [-WhatIf] [ <CommonParameters>]

Parameter Set: SnmpV3
Add-SCOMRunAsAccount [-Name] <String> [-UserName] <String> [-AuthProtocolAndKey
<PSCredential> ] [-ComputerName <String[]> ] [-Context <String> ] [-Credential
<PSCredential> ] [-Description <String> ] [-PrivacyProtocolAndKey <PSCredential> ] [-
SCSession <Connection[]> ] [-SnmpV3] [-Confirm] [-WhatIf] [ <CommonParameters>]

```

## Detailed Description

The Add-SCOMRunAsAccount cmdlet adds a Run As account to the management group.

By default all new accounts will be created with the more secure distribution option, with no approved systems. To modify the account distribution policy, use the Set-SCOMRunAsDistribution cmdlet.

## Parameters

### -ActionAccount

If set, the account will be an Action Account.

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

## **-AuthProtocolAndKey<PSCredential>**

The SNMP authentication protocol and key, stored in a PSCredential. Enter the protocol as the "username" and the key as the "password." Valid protocols are MD5 and SHA. Enter a PSCredential object, such as one that is returned by the Get-Credential cmdlet.

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

## **-Basic**

If set, the account will be a Basic runas account.

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

## **-Binary**

If set, the account will be a Binary runas account.

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

## **-CommunityString**

If set, the account will be a community string runas account.

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

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

Specifies a computer to establish a connection with. The computer must be running the System Center Data Access service. The default is the computer for the current management group connection.

Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

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

## **-Context<String>**

The SNMP context.

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

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

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

Specifies a user account under which the management group connection will run. It must have access to the server that is specified in the ComputerName parameter if specified. The default is the current user.

Enter a PSCredential object, such as one that is returned by the Get-Credential cmdlet.

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

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

The account description. If not set, will default to the display name.

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

### **-Digest**

If set, the account will be a Digest runas account.

Aliases	none
Required?	false
Position?	named
Default Value	none

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

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

The account name.

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

### **-Passphrase<SecureString>**

The SSH key passphrase used for cross-platform maintenance accounts.

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

### **-Path<String>**

The path to the binary data file or SSH key.

Aliases	none
Required?	true
Position?	2
Default Value	none
Accept Pipeline Input?	true (ByPropertyName)

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

### **-PrivacyProtocolAndKey<PSCredential>**

The SNMP privacy protocol and key, stored in a PSCredential. Enter the protocol as the "username" and the key as the "password." Valid protocols are AES and DES. Enter a PSCredential object, such as one that is returned by the Get-Credential cmdlet.

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

### **-Privileged**

If set, the cross-platform maintenance account will be have privileged access.

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

### **-RunAsCredential<PSCredential>**

The credential used for the runas account.

Aliases	none
Required?	true
Position?	2
Default Value	none
Accept Pipeline Input?	true (ByPropertyName)

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

### **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection.

Enter a Management group connection object, such as one returned by the Get-SCManagementGroupConnection cmdlet.

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

### **-SCXMaintenance**

If set, the account will be a cross-platform maintenance runas account.

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

### **-SCXMonitoring**

If set, the account will be a cross-platform monitoring runas account.

Aliases	none
Required?	false
Position?	named
Default Value	none



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

### **-Simple**

If set, the account will be a Simple runas account.

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

### **-SnmPV3**

If set, the account will be a SNMPV3 runas account.

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

### **-String<SecureString>**

The account community string.

Aliases	none
Required?	true
Position?	2
Default Value	none
Accept Pipeline Input?	true (ByPropertyName)

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

## **-Su**

If set, the cross-platform maintenance account will elevate using su to perform privileged actions.

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

## **-Sudo**

If set, the cross-platform account will elevate using sudo to perform privileged actions.

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

## **-SuPassword<SecureString>**

The SuperUser password used for a cross-platform maintenance account, when using su elevation.

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

## **-UserName<String>**

The account username. This parameter is only valid for SNMPV3 and cross-platform maintenance accounts, otherwise -RunAsCredential should be used.

Aliases	none
Required?	true
Position?	2
Default Value	none
Accept Pipeline Input?	true (ByPropertyName)
Accept Wildcard Characters?	false

## **-Windows**

If set, the account will be a Windows runas account. This is the 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 -----

This command adds a new Windows account.

```
PS C:\>Add-SCOMRunAsAccount -Windows -Name "Contoso.Windows" -DisplayName "Contoso domain account" -Description "Account used for monitoring the Contoso domain" -RunAsCredential (Get-Credential)
```

### ----- EXAMPLE 2 -----

This example adds a new community string account.

```
PS C:\>$communityString = Read-Host -AsSecureString
PS C:\>Add-SCOMRunAsAccount -CommunityString -Name "Contoso.CommStr" -String $communityString
```

### ----- EXAMPLE 3 -----

This command adds a new basic authentication account.

```
PS C:\>Add-SCOMRunAsAccount -Basic -Name "Contoso.Basic" -RunAsCredential (Get-Credential)
```

### ----- EXAMPLE 4 -----

This command adds a new simple authentication account.

```
PS C:\>Add-SCOMRunAsAccount -Simple -Name "Contoso.Simple" -RunAsCredential (Get-Credential)
```

## ----- EXAMPLE 5 -----

This command adds a new digest authentication account.

```
PS C:\>Add-SCOMRunAsAccount -Digest -Name "Contoso.Digest" -RunAsCredential (Get-Credential)
```

## ----- EXAMPLE 6 -----

This command adds a new binary account.

```
PS C:\>Add-SCOMRunAsAccount -Binary -Name "Contoso.Binary" -Path "C:\accountfile.bin"
```

## ----- EXAMPLE 7 -----

This command adds a new action account.

```
PS C:\>Add-SCOMRunAsAccount -ActionAccount -Name "Contoso.Action" -RunAsCredential (Get-Credential)
```

## ----- EXAMPLE 8 -----

This command adds a new SNMP v3 account with no context, authentication protocol, or privacy protocol.

```
PS C:\>Add-SCOMRunAsAccount -Snmv3 -Name "Contoso.Snmp1" -UserName "snmpuser"
```

## ----- EXAMPLE 9 -----

This example adds a new SNMP v3 account with context, authentication protocol, and privacy protocol specified.

```
PS C:\>$auth = Get-Credential
```

```
PS C:\>$privacy = Get-Credential
```

```
PS C:\>Add-SCOMRunAsAccount -Snmv3 -Name "Contoso.Snmp2" -UserName "snmpuser" -Context "snmp context" -AuthProtocolAndKey $auth -PrivacyProtocolAndKey $privacy
```

## ----- EXAMPLE 10 -----

This command adds a new SCX monitoring account, with sudo elevation.

```
PS C:\>Add-SCOMRunAsAccount -SCXMonitoring -Name "Contoso.SCXMon" -RunAsCredential (Get-Credential) -Sudo
```

## ----- EXAMPLE 11 -----

This example adds a new SCX maintenance account with privileged access, using a passphrase-protected SSH key.

```
PS C:\>$passphrase = Read-Host -AsSecureString
```

```
PS C:\>Add-SCOMRunAsAccount -SCXMaintenance -Name "Contoso.SCXMainSSH" -UserName "scxuser" -Path "C:\sshkey.ppk" -Passphrase $passphrase -Privileged
```

## ----- EXAMPLE 12 -----

This command adds a new SCX maintenance account without privileged access, using username and password, which uses sudo for elevation.

```
PS C:\>Add-SCOMRunAsAccount -SCXMaintenance -Name "Contoso.SCXMainUserName" -RunAsCredential (Get-Credential) -Sudo
```

## ----- EXAMPLE 13 -----

This example adds a new SCX maintenance account without privileged access, using username and password, which uses SU for elevation.

```
PS C:\>$suPassword = Read-Host -AsSecureString
```

```
PS C:\>Add-SCOMRunAsAccount -SCXMaintenance -Name "Contoso.SCXMainUserName" -RunAsCredential (Get-Credential) -Su -SuPassword $suPassword
```

# Add-SCOMRunAsProfile

---

## Add-SCOMRunAsProfile

Adds a new runas profile.

### Syntax

Parameter Set: Empty

```
Add-SCOMRunAsProfile [-Name] <String> [[-DisplayName] <String> ] [[-Description] <String> ]  
[[[-Comment] <String> ] [[-Guid] <Guid> ] -ManagementPack <ManagementPack[]> [-ComputerName  
<String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [-Confirm] [-WhatIf]  
[ <CommonParameters>]
```

### Detailed Description

The Add-SCOMRunAsProfile cmdlet adds a new Run as profile.

### Parameters

#### **-Comment<String>**

An administrative comment for the profile

Aliases	none
Required?	false
Position?	4
Default Value	none
Accept Pipeline Input?	true (ByPropertyName)
Accept Wildcard Characters?	false

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

Specifies a computer to establish a connection with. The computer must be running the System Center Data Access service. The default is the computer for the current management group connection.

Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

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

**-Credential<PSCredential>**

Specifies a user account under which the management group connection will run. It must have access to the server that is specified in the ComputerName parameter if specified. The default is the current user.

Enter a PSCredential object, such as one that is returned by the Get-Credential cmdlet.

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

**-Description<String>**

The description of the profile. Will use the displayname by default.

Aliases	none
Required?	false
Position?	3
Default Value	none
Accept Pipeline Input?	true (ByPropertyName)
Accept Wildcard Characters?	false

**-DisplayName<String>**

The display name of the profile. Will use the name by default.



Aliases	none
Required?	false
Position?	2
Default Value	none
Accept Pipeline Input?	true (ByPropertyName)
Accept Wildcard Characters?	false

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

The guid to use for the profile. If not set, a new GUID will be generated.

Aliases	none
Required?	false
Position?	4
Default Value	none
Accept Pipeline Input?	true (ByPropertyName)
Accept Wildcard Characters?	false

### **-ManagementPack<ManagementPack[]>**

The management pack in which the runas profile will be saved.

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

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

The name of the profile.

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

### **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection.

Enter a Management group connection object, such as one returned by the Get-SCManagementGroupConnection cmdlet.

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

### **-Confirm**

Prompts you for confirmation before executing the command.

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

### **-WhatIf**

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

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

## <CommonParameters>

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

## Examples

### ----- EXAMPLE 1 -----

This example adds a new runas profile to the default user management pack.

```
PS C:\>$mp = Get-SCOMManagementPack -Name "*DefaultUser"
```

```
PS C:\>Add-SCOMRunAsProfile -Name "Contoso.MonitoringProfile" -ManagementPack $mp
```

# Add-SCOMSubscriberSchedule

---

## Add-SCOMSubscriberSchedule

Adds a new notification subscriber schedule entry.

### Syntax

Parameter Set: AllDay

```
Add-SCOMSubscriberSchedule [[-DayOfWeek] <NotificationRecipientScheduleEntryDaysOfWeek> ] -  
AllDay-Subscriber <NotificationRecipient> [-ComputerName <String[]> ] [-Credential  
<PSCredential> ] [-EndDate <DateTime> ] [-Exclude] [-PassThru] [-SCSession <Connection[]> ]  
[-StartDate <DateTime> ] [-TimeZone <String> ] [ <CommonParameters>]
```

Parameter Set: StartAndEnd

```
Add-SCOMSubscriberSchedule [-StartTime] <DateTime> [-EndTime] <DateTime> [[-DayOfWeek]  
<NotificationRecipientScheduleEntryDaysOfWeek> ] -Subscriber <NotificationRecipient> [-  
ComputerName <String[]> ] [-Credential <PSCredential> ] [-EndDate <DateTime> ] [-Exclude] [-  
PassThru] [-SCSession <Connection[]> ] [-StartDate <DateTime> ] [-TimeZone <String> ] [  
<CommonParameters>]
```

### Detailed Description

The Add-SCOMSubscriberSchedule cmdlet adds a new notification subscriber schedule entry.

If no entries are added to a subscriber's schedule, they will always receive notifications.

### Parameters

#### -AllDay

If set, the schedule entry will be for the whole day.

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

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

Specifies a computer to establish a connection with. The computer must be running the System Center Data Access service. The default is the computer for the current management group connection.

Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

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

## **-Credential<PSCredential>**

Specifies a user account under which the management group connection will run. It must have access to the server that is specified in the ComputerName parameter if specified. The default is the current user.

Enter a PSCredential object, such as one that is returned by the Get-Credential cmdlet.

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

## **-DayOfWeek<NotificationRecipientScheduleEntryDaysOfWeek>**

Specifies the days of the week that the schedule entry is valid. By default the entry applies to all days of the week.

Aliases	none
Required?	false
Position?	3
Default Value	none

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

### **-EndDate<DateTime>**

The date to end the schedule entry. If no StartDate and EndDate are specified, the schedule entry will apply to all dates.

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

### **-EndTime<DateTime>**

The end time of the day for the schedule entry.

Aliases	none
Required?	true
Position?	2
Default Value	none
Accept Pipeline Input?	true (ByPropertyName)
Accept Wildcard Characters?	false

### **-Exclude**

If set, the times provided will be excluded from the schedule. Otherwise, the schedule will only include the provided times.

Aliases	none
Required?	false
Position?	named
Default Value	none

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

### **-PassThru**

If set, will pass though the subscriber object. In this way, many Add-SCOMSsubscriberSchedule commands can be used in the same pipeline.

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

### **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection.

Enter a Management group connection object, such as one returned by the Get-SCManagementGroupConnection cmdlet.

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

### **-StartDate<DateTime>**

The date to start the schedule entry. If no StartDate and EndDate are specified, the schedule entry will apply to all dates.

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

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

### **-StartTime<DateTime>**

The start time of day for the schedule entry.

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

### **-Subscriber<NotificationRecipient>**

The notification subscriber. To get a notification subscriber object, use Get-SCOMNotificationSubscriber or create a new one with Add-SCOMNotificationSubscriber.

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

### **-TimeZone<String>**

The time zone for the schedule entry. Defaults to the current user time zone. This parameter will do a wildcard match against TimeZone display names.

Examples of valid values are:

"\*UTC+05:00\*" United States Eastern time



"\*Pacific Time\*" United States Pacific time  
"\*Amsterdam\*" Amsterdam, Berlin, Bern, Rome, Stockholm time

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 -----

This example adds a schedule window from 9 AM to 5 PM on Mondays, Wednesdays, and Fridays in all date ranges to notification subscriber Katarina.

```
PS C:\>Get-SCOMNotificationSubscriber "Katarina" | Add-SCOMSubscriberSchedule -StartTime "9:00 AM" -EndTime "5:00 PM" -DayOfWeek Monday, Wednesday, Friday
```

### ----- EXAMPLE 2 -----

This example adds two schedule entries to notification subscriber Cesar, in the USA central time zone.

-----

```
PS C:\>Get-SCOMNotificationSubscriber "Cesar" | Add-SCOMSubscriberSchedule -StartTime "7:00 AM" -EndTime "4:00 PM" -DayOfWeek Monday, Wednesday, Friday -TimeZone "*UTC-06:00*" -PassThru | Add-SCOMSubscriberSchedule -StartDate '2012/1/1' -EndDate '2012/1/15' -TimeZone "*UTC-06:00*" -Exclude
```

# Add-SCOMTierConnector

---

## Add-SCOMTierConnector

Adds a connector to a management group tier.

### Syntax

Parameter Set: Default

```
Add-SCOMTierConnector -Connector <MonitoringConnector> -Tier <TieredManagementGroup> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-PassThru] [-SCSession <Connection[]> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The Add-SCOMTierConnector cmdlet adds a connector to a management group tier. Connectors are used to communicate monitoring information to another system.

### Parameters

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

Specifies a computer to establish a connection with. The computer must be running the System Center Data Access service. The default is the computer for the current management group connection.

Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

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

#### **-Connector<MonitoringConnector>**

Specifies the Operations Manager connector.

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

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

Specifies a user account under which the management group connection will run. It must have access to the server that is specified in the ComputerName parameter if specified. The default is the current user.

Enter a PSCredential object, such as one that is returned by the Get-Credential cmdlet.

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

### **-PassThru**

Passes the tier object created by this cmdlet through the pipeline. By default, this cmdlet does not pass any objects through the pipeline.

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

## **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection.

Enter a Management group connection object, such as one returned by the Get-SCManagementGroupConnection cmdlet.

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

## **-Tier<TieredManagementGroup>**

Specifies the Operations Manager tiered management group.

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 -----

This example adds a new connector to an existing tiered management group by using the pipe operator.

```
PS C:\>$tier = Get-SCOMTieredManagementGroup | Select-Object -First 1
```

```
PS C:\>Add-SCOMConnector -Name MyProductConnector | Add-SCOMTierConnector -Tier $tier
```

# Add-SCOMTieredManagementGroup

---

## Add-SCOMTieredManagementGroup

Adds a tiered management group to the management group.

### Syntax

Parameter Set: SimpleAccountTier

```
Add-SCOMTieredManagementGroup -ConnectionCredential <PSCredential> -Name <String> -  
ServerName <String> [-CacheConfiguration <CacheConfiguration> ] [-CacheMode <CacheMode> ] [-  
ComputerName <String[]> ] [-Credential <PSCredential> ] [-InactivityTimeout <TimeSpan> ] [-  
SCSession <Connection[]> ] [-SendReceiveTimeout <TimeSpan> ] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: RunAsAccountTier

```
Add-SCOMTieredManagementGroup -ConnectionCredential <PSCredential> -Name <String> -  
RunAsAccount <WindowsCredentialSecureData> -ServerName <String> [-AvailableForConnectors] [-  
CacheConfiguration <CacheConfiguration> ] [-CacheMode <CacheMode> ] [-ComputerName  
<String[]> ] [-Credential <PSCredential> ] [-InactivityTimeout <TimeSpan> ] [-SCSession  
<Connection[]> ] [-SendReceiveTimeout <TimeSpan> ] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

### Detailed Description

The Add-SCOMTieredManagementGroup cmdlet adds a tiered management group to the management group.

### Parameters

#### -AvailableForConnectors

If set, the new tier will be available for connectors.

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

## **-CacheConfiguration<CacheConfiguration>**

The cache configuration used for the tiered management group.

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

## **-CacheMode<CacheMode>**

The cache mode used for the tiered management group.

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

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

Specifies a computer to establish a connection with. The computer must be running the System Center Data Access service. The default is the computer for the current management group connection.

Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

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

## **-ConnectionCredential<PSCredential>**

The credential used to connect to the tiered management group SDK server.

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

## **-Credential<PSCredential>**

Specifies a user account under which the management group connection will run. It must have access to the server that is specified in the ComputerName parameter if specified. The default is the current user.

Enter a PSCredential object, such as one that is returned by the Get-Credential cmdlet.

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

## **-InactivityTimeout<TimeSpan>**

The inactivity timeout used for the tiered management group.

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



## **-Name<String>**

The name of the new tiered management group.

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

## **-RunAsAccount<WindowsCredentialSecureData>**

The runas credential used when the tier is available for connectors.

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

## **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection.

Enter a Management group connection object, such as one returned by the Get-SCManagementGroupConnection cmdlet.

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

## **-SendReceiveTimeout<TimeSpan>**

The send/receive timeout used for the tiered management group.

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

## **-ServerName<String>**

The name of the SDK server to connect to in the tiered management group.

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 -----

This command adds a new tiered management group.

```
PS C:\>Add-SCOMTieredManagementGroup -Name "New tier" -ServerName "SCOM02.contoso.com" -ConnectionCredential (Get-Credential)
```

### ----- EXAMPLE 2 -----

This command adds a new tiered management group which is available for connectors.

```
PS C:\>Add-SCOMTieredManagementGroup -Name "New tier for connectors" -ServerName "SCOM02.contoso.com" -ConnectionCredential (Get-Credential) -AvailableForConnectors -RunAsAccount (Get-SCOMRunAsAccount "tieraccount")
```

# Add-SCOMUserRole

---

## Add-SCOMUserRole

Adds a new user role to the management group.

### Syntax

Parameter Set: Operator

```
Add-SCOMUserRole [-Name] <String> [[-DisplayName] <String> ] [[-Description] <String> ] [[-Users] <String[]> ] -Operator[-ComputerName <String[]> ] [-Credential <PSCredential> ] [-GroupScope <MonitoringObjectGroup[]> ] [-SCSession <Connection[]> ] [-TaskScope <ManagementPackTask[]> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: AdvancedOperator

```
Add-SCOMUserRole [-Name] <String> [[-DisplayName] <String> ] [[-Description] <String> ] [[-Users] <String[]> ] -AdvancedOperator[-ComputerName <String[]> ] [-Credential <PSCredential> ] [-GroupScope <MonitoringObjectGroup[]> ] [-SCSession <Connection[]> ] [-TaskScope <ManagementPackTask[]> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: Author

```
Add-SCOMUserRole [-Name] <String> [[-DisplayName] <String> ] [[-Description] <String> ] [[-Users] <String[]> ] -Author[-ClassScope <ManagementPackClass[]> ] [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-GroupScope <MonitoringObjectGroup[]> ] [-SCSession <Connection[]> ] [-TaskScope <ManagementPackTask[]> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ReadOnlyOperator

```
Add-SCOMUserRole [-Name] <String> [[-DisplayName] <String> ] [[-Description] <String> ] [[-Users] <String[]> ] -ReadOnlyOperator[-ComputerName <String[]> ] [-Credential <PSCredential> ] [-GroupScope <MonitoringObjectGroup[]> ] [-SCSession <Connection[]> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The Add-SCOMUserRole cmdlet adds a new user role to the management group.

### Parameters

#### -AdvancedOperator

Indicates that the new role is an Advanced Operator.

Aliases	none
Required?	true
Position?	named

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

## **-Author**

Indicates that the new role is an Author.

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

## **-ClassScope<ManagementPackClass[]>**

Specifies the classes that an Author role has access to. If this parameter is left unspecified, the role will be scoped to all classes. To indicate that no classes are in scope, specify \$null or an empty array @().

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

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

Specifies a computer to establish a connection with. The computer must be running the System Center Data Access service. The default is the computer for the current management group connection.

Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

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

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

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

Specifies a user account under which the management group connection will run. It must have access to the server that is specified in the ComputerName parameter if specified. The default is the current user.

Enter a PSCredential object, such as one that is returned by the Get-Credential cmdlet.

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

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

Specifies the description of the user role. By default, displayname is used.

Aliases	none
Required?	false
Position?	3
Default Value	none
Accept Pipeline Input?	true (ByPropertyName)
Accept Wildcard Characters?	false

### **-DisplayName<String>**

Specifies the display name of the user role. By default, name is used.

Aliases	none
Required?	false
Position?	2
Default Value	none
Accept Pipeline Input?	true (ByPropertyName)
Accept Wildcard Characters?	false

### **-GroupScope<MonitoringObjectGroup[]>**

Specifies the groups that a user role has access to. If this parameter is left unspecified, the role is scoped to all groups. To indicate that no groups are in scope, specify \$null or an empty array @().

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 user role.

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

### **-Operator**

Indicates that the new role is an Operator.

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

### **-ReadOnlyOperator**

Indicates that the new role is a Read-Only Operator.

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

### **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection.

Enter a Management group connection object, such as one returned by the Get-SCManagementGroupConnection cmdlet.

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



## **-TaskScope<ManagementPackTask[]>**

Specifies the tasks that an Author, Operator, or Advanced Operator role has access to. If this parameter is left unspecified, the role is scoped to all tasks. To indicate that no tasks are in scope, specify \$null or an empty array @().

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

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

A list of user names which are part of this user role.

Aliases	none
Required?	false
Position?	4
Default Value	none
Accept Pipeline Input?	true (ByPropertyName)
Accept Wildcard Characters?	false

## **-Confirm**

Prompts you for confirmation before executing the command.

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

## -WhatIf

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

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

## <CommonParameters>

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

## Examples

### ----- EXAMPLE 1 -----

This example adds a new read-only operator user role.

```
PS C:\>Add-SCOMUserRole -Name 'Script Role' -ReadOnlyOperator
```

### ----- EXAMPLE 2 -----

The first two commands get all task and group objects that have SQL in their name and store the objects in the \$approvedTasks and \$approvedGroups variables, respectively.

The last command creates a user role containing the users Kararina and Cesar that is scoped to the tasks and groups stored in \$approvedTasks and \$approvedGroups.

```
PS C:\> $approvedTasks = Get-SCOMTask -Name '*SQL*'
```

```
PS C:\>$approvedGroups = Get-SCOMGroup -Name '*SQL*'
```

```
PS C:\>Add-SCOMUserRole -Name 'SQL Operator' -Operator -GroupScope $approvedGroups -  
TaskScope $approvedTasks -User 'contoso\Katarina','contoso\Cesar'
```

# Approve-SCOMPendingManagement

---

## Approve-SCOMPendingManagement

Approves the specified pending agent management actions.

### Syntax

Parameter Set: FromAgentPendingAction

```
Approve-SCOMPendingManagement [-PendingAction] <AgentPendingAction[]> [[-ActionAccount] <PSCredential> ] [-PassThru] [ <CommonParameters>]
```

### Detailed Description

The Approve-SCOMPendingManagement cmdlet approves the specified pending management actions.

### Parameters

#### **-ActionAccount<PSCredential>**

Specifies the credentials under which the pending action will be approved.

Aliases	none
Required?	false
Position?	2
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

#### **-PassThru**

Returns an object representing the pending action. By default, this cmdlet does not generate any output.

Aliases	none
Required?	false
Position?	named

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

## **-PendingAction<AgentPendingAction[]>**

Specifies one or more pending actions to approve. Enter a variable that represents the pending actions, or type a command that gets the pending actions.

For information about how to get a pending action object, type Get-Help Get-SCOMPendingManagement.

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

## **<CommonParameters>**

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

## **Inputs**

The input type is the type of the objects that you can pipe to the cmdlet.

- **Microsoft.EnterpriseManagement.Administration.AgentPendingAction** Represents a task that targets an agent on a managed computer. The task is queued or awaiting administrator approval.

## **Examples**

### **----- EXAMPLE 1 -----**

Description

-----

This command gets the agent management entries that are pending with an action of "ManualApproval", and, by using the WhatIf parameter, displays what action would occur if the command was implemented. In this case, all targets with a pending action of "ManualApproval" would be approved.

```
PS C:\>Get-SCOMPendingManagement | where {$_.AgentPendingActionType -eq "ManualApproval"} |  
Approve-SCOMPendingManagement -WhatIf
```

## **Related topics**

[Deny-SCOMPendingManagement](#)

[Get-SCOMPendingManagement](#)

# Clear-SCOMSubscriberSchedule

---

## Clear-SCOMSubscriberSchedule

Removes all entries in a notification subscriber's schedule.

### Syntax

Parameter Set: Default

```
Clear-SCOMSubscriberSchedule [-Subscriber] <NotificationRecipient> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-PassThru] [-SCSession <Connection[]> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The Clear-SCOMSubscriberSchedule cmdlet removes all entries in a notification subscriber's schedule. A clear schedule indicates an "always on" subscription that will be notified at any time.

### Parameters

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

Specifies a computer to establish a connection with. The computer must be running the System Center Data Access service. The default is the computer for the current management group connection.

Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

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

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

Specifies a user account under which the management group connection will run. It must have access to the server that is specified in the ComputerName parameter if specified. The default is the current user.

Enter a PSCredential object, such as one that is returned by the Get-Credential cmdlet.

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

## **-PassThru**

Specifies a connection to a management server. The default is the current management group connection.

Enter a Management group connection object, such as one returned by the Get-SCManagementGroupConnection cmdlet.

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

## **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection.

Enter a Management group connection object, such as one returned by the Get-SCManagementGroupConnection cmdlet.

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

## **-Subscriber<NotificationRecipient>**

Specifies a connection to a management server. The default is the current management group connection.

Enter a Management group connection object, such as one returned by the Get-SCManagementGroupConnection cmdlet.

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

## **-Confirm**

Prompts you for confirmation before executing the command.

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

## **-WhatIf**

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

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



## <CommonParameters>

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

## Examples

### ----- EXAMPLE 1 -----

This example resets the schedules of all subscribers

```
PS C:\>Get-SCOMNotificationSubscriber | Clear-SCOMSubscriberSchedule
```

# Deny-SCOMPendingManagement

---

## Deny-SCOMPendingManagement

Denies the specified pending agent management actions.

### Syntax

Parameter Set: FromAgentPendingAction

```
Deny-SCOMPendingManagement [-PendingAction] <AgentPendingAction[]> [-PassThru] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The Deny-SCOMPendingManagement cmdlet denies the specified pending management actions.

### Parameters

#### -PassThru

Returns an object representing the pending action. By default, this cmdlet does not generate any output.

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

#### -PendingAction<AgentPendingAction[]>

Specifies one or more pending actions to deny. Enter a variable that represents the pending actions, or type a command that gets the pending actions.

For information about how to get a pending action object, type `Get-Help Get-SCOMPendingManagement`.

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

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

## **-Confirm**

Prompts you for confirmation before executing the command.

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

## **-WhatIf**

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

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

## **<CommonParameters>**

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

## **Inputs**

The input type is the type of the objects that you can pipe to the cmdlet.

- **Microsoft.EnterpriseManagement.Administration.AgentPendingAction** Represents a task that targets an agent on a managed computer. The task is queued or awaiting administrator approval.

## Examples

### ----- EXAMPLE 1 -----

Description

-----

This command gets the agent management entries that are pending with an action of "ManualApproval", and, by using the *WhatIf* parameter, displays what action would occur if the command was implemented. In this case, all targets with a pending action of "ManualApproval" would be denied.

```
PS C:\>Get-SCOMPendingManagement | where {$_.AgentPendingActionType -eq "ManualApproval"} |  
Deny-SCOMPendingManagement -WhatIf
```

## Related topics

[Approve-SCOMPendingManagement](#)

[Get-SCOMPendingManagement](#)

# Disable-SCOMAgentProxy

---

## Disable-SCOMAgentProxy

Disables agents from acting as a proxy and discovering managed objects on other computers.

### Syntax

Parameter Set: FromAgent

```
Disable-SCOMAgentProxy [-Agent] <AgentManagedComputer[]> [[-PassThru]] [-Confirm] [-WhatIf]  
[ <CommonParameters>]
```

### Detailed Description

The Disable-SCOMAgentProxy cmdlet disables agents from acting as a proxy and discovering managed objects on other computers. This cmdlet requires an agent object.

### Parameters

#### -Agent<AgentManagedComputer[]>

Specifies one or more agent objects. Enter a variable that represents the agents, or type a command that gets the agents. For information about how to get an agent object, type Get-Help Get-SCOMAgent.

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

#### -PassThru

Returns an object representing the updated settings. By default, this cmdlet does not generate any output.

Aliases	none
Required?	false

Position?	2
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 -----

Description

-----

This command gets the agent named "Server01.Contoso.com" and disables it from acting as a proxy.

```
PS C:\>"Server01.Contoso.com" | Get-SCOMAgent | Disable-SCOMAgentProxy
```

## ----- EXAMPLE 2 -----

Description

-----

This command gets all agents that have their ProxyingEnabled property set to True and then disables the agents from acting as a proxy.

```
PS C:\>Get-SCOMAgent | where {$_.ProxyingEnabled.Value -eq $True} | Disable-SCOMAgentProxy
```

### Related topics

[Enable-SCOMAgentProxy](#)

[Get-SCOMAgent](#)

# Disable-SCOMDiscovery

---

## Disable-SCOMDiscovery

Disables discoveries by creating and saving overrides for the specified discoveries.

### Syntax

Parameter Set: Empty

```
Disable-SCOMDiscovery [-ManagementPack] <ManagementPack> [-Discovery]  
<ManagementPackDiscovery[]> [[-Enforce]] [[-PassThru]] [ <CommonParameters>]
```

Parameter Set: FromGroup

```
Disable-SCOMDiscovery [[-Group] <MonitoringObject[]> ] [-ManagementPack] <ManagementPack> [-  
Discovery] <ManagementPackDiscovery[]> [[-Enforce]] [[-PassThru]] [ <CommonParameters>]
```

Parameter Set: FromInstance

```
Disable-SCOMDiscovery [[-Instance] <MonitoringObject[]> ] [-ManagementPack] <ManagementPack>  
[-Discovery] <ManagementPackDiscovery[]> [[-Enforce]] [[-PassThru]] [ <CommonParameters>]
```

Parameter Set: FromManagementPackClass

```
Disable-SCOMDiscovery [-Class] <ManagementPackClass[]> [-ManagementPack] <ManagementPack> [-  
Discovery] <ManagementPackDiscovery[]> [[-Enforce]] [[-PassThru]] [ <CommonParameters>]
```

### Detailed Description

The Disable-SCOMDiscovery cmdlet disables discoveries by creating and saving overrides for the specified discoveries.

### Parameters

#### **-Class<ManagementPackClass[]>**

Specifies one or more class objects for which the discoveries will be disabled. Enter a variable that represents the classes, or type a command that gets the classes. For information about how to get a class object, type `Get-Help Get-SCOMClass`.

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



## **-Discovery<ManagementPackDiscovery[]>**

Specifies one or more discovery objects. Enter a variable that represents the discoveries, or type a command that gets the discoveries.

For information about how to get a discovery object, type Get-Help Get-SCOMDiscovery.

Aliases	none
Required?	true
Position?	2
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-Enforce**

Indicates that the enforce property will be set to True on the override that is created to disable the specified discoveries.

Aliases	none
Required?	false
Position?	3
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-Group<MonitoringObject[]>**

Specifies one or more group objects for which the discoveries will be disabled. Enter a variable that represents the groups, or type a command that gets the groups. For information about how to get a group object, type Get-Help Get-SCOMGroup.

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

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

### **-Instance<MonitoringObject[]>**

Specifies one or more class instance objects for which the discoveries will be disabled. Enter a variable that represents the class instances, or type a command that gets the class instances. This parameter also accepts group objects. For information about how to get a class instance object, type Get-Help Get-SCOMClassInstance.

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

### **-ManagementPack<ManagementPack>**

Specifies one or more management pack objects into which the disabling override will be saved. If the discovery is contained in an unsealed management pack, the overrides must be saved into the same management pack. Enter a variable that represents the management packs, or type a command that gets the management packs. For information about how to get a management pack object, type Get-Help Get-SCOMManagementPack.

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

### **-PassThru**

Returns an object representing the discovery. By default, this cmdlet does not generate any output.

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

Required?	false
Position?	4
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## <CommonParameters>

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

## Examples

### ----- EXAMPLE 1 -----

Description

-----

The first three commands get a management pack object, a class object, and a discovery object, and then store the objects in the \$MP, \$Class, and \$Discovery variables, respectively.

The last command disables the discovery \$Discovery for the management pack \$MP.

```
PS C:\>$MP = Get-SCOMManagementPack -displayname "My SQL MP Customization" | where
{$_ .Sealed -eq $False}
```

```
PS C:\>$Class = Get-SCOMClass -DisplayName "SQL DB Engine"
```

```
PS C:\>$Discovery = Get-SCOMDiscovery -DisplayName *rule*
```

```
PS C:\>Disable-SCOMDiscovery -Class $Class -Discovery $Discovery -ManagementPack $MP
```

## Related topics

[Get-SCOMClass](#)

[Get-SCOMClassInstance](#)

[Get-SCOMDiscovery](#)

[Get-SCOMManagementPack](#)

[Enable-SCOMDiscovery](#)

# Disable-SCOMMonitor

---

## Disable-SCOMMonitor

Disables monitors by creating and saving overrides for the specified monitors.

### Syntax

Parameter Set: Empty

```
Disable-SCOMMonitor [-ManagementPack] <ManagementPack> [-Monitor] <ManagementPackMonitor[]> [[-Enforce]] [[-PassThru]] [ <CommonParameters>]
```

Parameter Set: FromGroup

```
Disable-SCOMMonitor [[-Group] <MonitoringObject[]> ] [-ManagementPack] <ManagementPack> [-Monitor] <ManagementPackMonitor[]> [[-Enforce]] [[-PassThru]] [ <CommonParameters>]
```

Parameter Set: FromInstance

```
Disable-SCOMMonitor [[-Instance] <MonitoringObject[]> ] [-ManagementPack] <ManagementPack> [-Monitor] <ManagementPackMonitor[]> [[-Enforce]] [[-PassThru]] [ <CommonParameters>]
```

Parameter Set: FromManagementPackClass

```
Disable-SCOMMonitor [-Class] <ManagementPackClass[]> [-ManagementPack] <ManagementPack> [-Monitor] <ManagementPackMonitor[]> [[-Enforce]] [[-PassThru]] [ <CommonParameters>]
```

### Detailed Description

The Disable-SCOMMonitor cmdlet disables monitors by creating and saving overrides for the specified monitors.

### Parameters

#### **-Class<ManagementPackClass[]>**

Specifies one or more class objects for which the monitors will be disabled. Enter a variable that represents the classes, or type a command that gets the classes. For information about how to get a class object, type Get-Help Get-SCOMClass.

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

## **-Enforce**

Indicates that the enforce property will be set to True on the override that is created to disable the specified monitors.

Aliases	none
Required?	false
Position?	3
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-Group<MonitoringObject[]>**

Specifies one or more group objects for which the monitors will be disabled. Enter a variable that represents the groups, or type a command that gets the groups. For information about how to get a group object, type Get-Help Get-SCOMGroup.

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

## **-Instance<MonitoringObject[]>**

Specifies one or more class instance objects for which the monitors will be disabled. Enter a variable that represents the class instances, or type a command that gets the class instances. This parameter also accepts group objects. For information about how to get a class instance object, type Get-Help Get-SCOMClassInstance.

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

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

### **-ManagementPack<ManagementPack>**

Specifies one or more management pack objects into which the disabling override will be saved. If the monitor is contained in an unsealed management pack, the overrides must be saved into the same management pack. Enter a variable that represents the management packs, or type a command that gets the management packs. For information about how to get a management pack object, type Get-Help Get-SCOMManagementPack.

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

### **-Monitor<ManagementPackMonitor[]>**

Specifies one or more monitor objects. Enter a variable that represents the monitors, or type a command that gets the monitors.

For information about how to get a monitor object, type Get-Help Get-SCOMMonitor.

Aliases	none
Required?	true
Position?	2
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-PassThru**

Returns an object representing the monitor. By default, this cmdlet does not generate any output.

Aliases	none
Required?	false

Position?	4
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## <CommonParameters>

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

## Examples

### ----- EXAMPLE 1 -----

Description

-----

The first three commands get a management pack object, a class object, and a monitor object, and then store the objects in the \$MP, \$Class, and \$Monitor variables, respectively.

The fourth command disables the monitor stored in \$Monitor for all instances of the class stored in \$Class using Enforce property. The override is stored in the management pack stored in \$MP.

```
PS C:\>$MP = Get-SCOMManagementPack -displayname "My SQL MP Customization" | where
{$_ .Sealed -eq $False}
PS C:\>$Class = Get-SCOMClass -DisplayName "SQL DB Engine"
PS C:\>$Monitor = Get-SCOMMonitor -DisplayName *memory*
PS C:\>Disable-SCOMMonitor -Class $Class -ManagementPack $MP -Monitor $Monitor -Enforce
```

## Related topics

[Get-SCOMClass](#)

[Get-SCOMClassInstance](#)

[Get-SCOMManagementPack](#)

[Enable-SCOMMonitor](#)

[Get-SCOMMonitor](#)

# Disable-SCOMNotificationSubscription

---

## Disable-SCOMNotificationSubscription

This cmdlet disables a notification subscription.

### Syntax

Parameter Set: Default

```
Disable-SCOMNotificationSubscription [-Subscription] <NotificationSubscription[]> [-PassThru] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Parameters

#### -PassThru

Returns an object representing the notification subscription. By default, this cmdlet does not generate any output.

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

#### -Subscription<NotificationSubscription[]>

Specifies which subscriptions to change.

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



## -Confirm

Prompts you for confirmation before executing the command.

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

## -WhatIf

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

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

## <CommonParameters>

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

## Examples

### ----- EXAMPLE 1 -----

This command disables all enabled notifications.

```
PS C:\> Get-SCOMNotificationSubscription | where-object{$_ .Enabled} | Disable-SCOMNotificationSubscription
```

### ----- EXAMPLE 2 -----

This command disables all notifications where the user CONTOSO\AIYoung is on the TO line.

```
PS C:\> Get-SCOMNotificationSubscription | where-object{$_ .ToRecipients -contains  
"CONTOSO\A1Young"} | Disable-SCOMNotificationSubscription
```

# Disable-SCOMOperationalDataReporting

---

## Disable-SCOMOperationalDataReporting

Disables operational data reporting for the management group.

### Syntax

Parameter Set: Default

```
Disable-SCOMOperationalDataReporting [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

### Detailed Description

The Disable-SCOMOperationalDataReporting cmdlet disables operational data reporting for the management group.

### Parameters

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

Specifies a computer to establish a connection with. The computer must be running the System Center Data Access service. The default is the computer for the current management group connection.

Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

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

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

Specifies a user account under which the management group connection will run. It must have access to the server that is specified in the ComputerName parameter if specified. The default is the current user.

Enter a PSCredential object, such as one that is returned by the Get-Credential cmdlet.

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

### **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection.

Enter a Management group connection object, such as one returned by the Get-SCManagementGroupConnection cmdlet.

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

### **<CommonParameters>**

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

### **Examples**

#### **----- EXAMPLE 1 -----**

This example disables operational data reporting.

```
PS C:\>Disable-SCOMOperationalDataReporting
```

# Disable-SCOMRule

---

## Disable-SCOMRule

Disables monitoring rules by creating and saving overrides for the specified rules.

### Syntax

Parameter Set: Empty

```
Disable-SCOMRule [-ManagementPack] <ManagementPack> [-Rule] <ManagementPackRule[]> [[-Enforce]] [[-PassThru]] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: FromGroup

```
Disable-SCOMRule [[-Group] <MonitoringObject[]> ] [-ManagementPack] <ManagementPack> [-Rule] <ManagementPackRule[]> [[-Enforce]] [[-PassThru]] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: FromInstance

```
Disable-SCOMRule [[-Instance] <MonitoringObject[]> ] [-ManagementPack] <ManagementPack> [-Rule] <ManagementPackRule[]> [[-Enforce]] [[-PassThru]] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: FromManagementPackClass

```
Disable-SCOMRule [-Class] <ManagementPackClass[]> [-ManagementPack] <ManagementPack> [-Rule] <ManagementPackRule[]> [[-Enforce]] [[-PassThru]] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The Disable-SCOMRule cmdlet disables monitoring rules by creating and saving overrides for the specified rules.

### Parameters

#### **-Class<ManagementPackClass[]>**

Specifies one or more class objects for which the rules will be disabled. Enter a variable that represents the classes, or type a command that gets the classes. For information about how to get a class object, type Get-Help Get-SCOMClass.

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

## **-Enforce**

Indicates that the enforce property will be set to True on the override that is created to disable the specified monitoring rules.

Aliases	none
Required?	false
Position?	3
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-Group<MonitoringObject[]>**

Specifies one or more group objects for which the rules will be disabled. Enter a variable that represents the groups, or type a command that gets the groups. For information about how to get a group object, type Get-Help Get-SCOMGroup.

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

## **-Instance<MonitoringObject[]>**

Specifies one or more class instance objects for which the rules will be disabled. Enter a variable that represents the class instances, or type a command that gets the class instances. This parameter also accepts group objects. For information about how to get a class instance object, type Get-Help Get-SCOMClassInstance.

Aliases	none
Required?	false
Position?	1

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

### **-ManagementPack<ManagementPack>**

Specifies one or more management pack objects into which the disabling override will be saved. If the rule is contained in an unsealed management pack, the overrides must be saved into the same management pack. Enter a variable that represents the management packs, or type a command that gets the management packs. For information about how to get a management pack object, type Get-Help Get-SCOMManagementPack.

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

### **-PassThru**

Returns an object representing the monitoring rule. By default, this cmdlet does not generate any output.

Aliases	none
Required?	false
Position?	4
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-Rule<ManagementPackRule[]>**

Specifies one or more monitoring rule objects. Enter a variable that represents the monitoring rules, or type a command that gets the monitoring rules.

For information about how to get a monitoring rule object, type Get-Help Get-SCOMRule.

Aliases	none
Required?	true
Position?	2
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-Confirm**

Prompts you for confirmation before executing the command.

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

## **-WhatIf**

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

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

## **<CommonParameters>**

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



## Examples

### ----- EXAMPLE 1 -----

Description

-----

The first three commands get a management pack object, a class object, and a monitoring rule object, and then store the objects in the \$MP, \$Class, and \$Rule variables, respectively.

The last command disables the monitoring rule stored in \$Rule for the management pack stored in \$MP using the –Enforce property.

```
PS C:\>$MP = Get-SCOMManagementPack -displayname "My SQL MP Customization" | where  
{$_Sealed -eq $False}
```

```
PS C:\>$Class = Get-SCOMClass -DisplayName "SQL DB Engine"
```

```
PS C:\>$Rule = Get-SCOMRule -DisplayName "*Events/sec"
```

```
PS C:\>Disable-SCOMRule -Class $Class -Rule $Rule -ManagementPack $MP -Enforce
```

## Related topics

[Get-SCOMClass](#)

[Get-SCOMClassInstance](#)

[Get-SCOMManagementPack](#)

[Enable-SCOMRule](#)

[Get-SCOMRule](#)

# Enable-SCOMAgentProxy

---

## Enable-SCOMAgentProxy

Enables agents to act as a proxy and discover managed objects on other computers.

### Syntax

Parameter Set: FromAgent

```
Enable-SCOMAgentProxy [-Agent] <AgentManagedComputer[]> [[-PassThru]] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

### Detailed Description

The Enable-SCOMAgentProxy cmdlet enables agents to act as a proxy and discover managed objects on other computers. This cmdlet requires an agent object.

### Parameters

#### **-Agent<AgentManagedComputer[]>**

Specifies one or more agent objects. Enter a variable that represents the agents, or type a command that gets the agents. For information about how to get an agent object, type Get-Help Get-SCOMAgent.

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

#### **-PassThru**

Returns an object representing the updated settings. By default, this cmdlet does not generate any output.

Aliases	none
Required?	false

Position?	2
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 -----

Description

-----

This command gets the agent named "Server01.Contoso.com" and enables the agent to act as a proxy for other agents.

```
PS C:\>"Server01.Contoso.com" | Get-SCOMAgent | Enable-SCOMAgentProxy -PassThru
```

## ----- EXAMPLE 2 -----

Description

-----

This command gets all agents that have their ProxyingEnabled value set to false and then enables the agents to act as a proxy after confirming the action.

```
PS C:\>Get-SCOMAgent | where {$_.ProxyingEnabled.Value -eq $False} | Enable-SCOMAgentProxy -Confirm
```

## ----- EXAMPLE 3 -----

Description

-----

The Proxying Enabled value is stored in the ProxyingEnabled property of the object that represents an SCAgent. The first command gets the agent named "Server01.Contoso.com", and the second command displays the value for the ProxyingEnabled property of that agent.

```
PS C:\>$Agent = Get-SCOMAgent -Name "Server01.Contoso.com"
```

```
PS C:\>$Agent.ProxyingEnabled
```

## Related topics

[Disable-SCOMAgentProxy](#)

[Get-SCOMAgent](#)

# Enable-SCOMDiscovery

---

## Enable-SCOMDiscovery

Enables discoveries by creating and saving overrides for the specified discoveries.

### Syntax

Parameter Set: Empty

```
Enable-SCOMDiscovery [-ManagementPack] <ManagementPack> [-Discovery]  
<ManagementPackDiscovery[]> [[-Enforce]] [[-PassThru]] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: FromGroup

```
Enable-SCOMDiscovery [[-Group] <MonitoringObject[]> ] [-ManagementPack] <ManagementPack> [-  
Discovery] <ManagementPackDiscovery[]> [[-Enforce]] [[-PassThru]] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: FromInstance

```
Enable-SCOMDiscovery [[-Instance] <MonitoringObject[]> ] [-ManagementPack] <ManagementPack>  
[-Discovery] <ManagementPackDiscovery[]> [[-Enforce]] [[-PassThru]] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: FromManagementPackClass

```
Enable-SCOMDiscovery [-Class] <ManagementPackClass[]> [-ManagementPack] <ManagementPack> [-  
Discovery] <ManagementPackDiscovery[]> [[-Enforce]] [[-PassThru]] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

### Detailed Description

The Enable-Discovery cmdlet enables discoveries by creating and saving overrides for the specified discoveries.

### Parameters

#### **-Class<ManagementPackClass[]>**

Specifies one or more class objects for which the discoveries will be enabled. Enter a variable that represents the classes, or type a command that gets the classes. For information about how to get a class object, type `Get-Help Get-SCOMClass`.

Aliases	none
Required?	true
Position?	1
Default Value	none

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

### **-Discovery<ManagementPackDiscovery[]>**

Specifies one or more discovery objects. Enter a variable that represents the discoveries, or type a command that gets the discoveries.

For information about how to get a discovery object, type Get-Help Get-SCOMDiscovery.

Aliases	none
Required?	true
Position?	2
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-Enforce**

Indicates that the enforce property will be set to True on the override that is created to enable the specified discoveries.

Aliases	none
Required?	false
Position?	3
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-Group<MonitoringObject[]>**

Specifies one or more group objects for which the discoveries will be enabled. Enter a variable that represents the groups, or type a command that gets the groups. For information about how to get a group object, type Get-Help Get-SCOMGroup.

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

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

### **-Instance<MonitoringObject[]>**

Specifies one or more class instance objects for which the discoveries will be enabled. Enter a variable that represents the class instances, or type a command that gets the class instances. This parameter also accepts group objects. For information about how to get a class instance object, type Get-Help Get-SCOMClassInstance.

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

### **-ManagementPack<ManagementPack>**

Specifies one or more management pack objects into which the enabling override will be saved. If the discovery is contained in an unsealed management pack, the overrides must be saved into the same management pack. Enter a variable that represents the management packs, or type a command that gets the management packs. For information about how to get a management pack object, type Get-Help Get-SCOMManagementPack.

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

## **-PassThru**

Returns an object representing the discovery. By default, this cmdlet does not generate any output.

Aliases	none
Required?	false
Position?	4
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-Confirm**

Prompts you for confirmation before executing the command.

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

## **-WhatIf**

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

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

## **<CommonParameters>**

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



## Examples

### ----- EXAMPLE 1 -----

Description

-----

The first three commands get a management pack object, a class object, and a discovery object, and then store the objects in the \$MP, \$Class, and \$Discovery variables, respectively.

The fourth command enables the discovery stored in \$Discovery for the management pack stored in \$MP using the Enforce property.

```
PS C:\>$MP = Get-SCOMManagementPack -displayname "My SQL MP Customization" | where  
{$_Sealed -eq $False}
```

```
PS C:\>$Class = Get-SCOMClass -DisplayName "SQL DB Engine"
```

```
PS C:\>$Discovery = Get-SCOMDiscovery -DisplayName *rule*
```

```
PS C:\>Enable-SCOMDiscovery -Class $Class -Discovery $Discovery -ManagementPack $MP -Enforce
```

## Related topics

[Get-SCOMClass](#)

[Get-SCOMClassInstance](#)

[Get-SCOMDiscovery](#)

[Get-SCOMGroup](#)

[Get-SCOMManagementPack](#)

[Disable-SCOMDiscovery](#)

# Enable-SCOMMonitor

---

## Enable-SCOMMonitor

Enables monitors by creating and saving overrides for the specified monitors.

### Syntax

Parameter Set: Empty

```
Enable-SCOMMonitor [-ManagementPack] <ManagementPack> [-Monitor] <ManagementPackMonitor[]>
[[-Enforce]] [[-PassThru]] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: FromGroup

```
Enable-SCOMMonitor [[-Group] <MonitoringObject[]> ] [-ManagementPack] <ManagementPack> [-
Monitor] <ManagementPackMonitor[]> [[-Enforce]] [[-PassThru]] [-Confirm] [-WhatIf] [
<CommonParameters>]
```

Parameter Set: FromInstance

```
Enable-SCOMMonitor [[-Instance] <MonitoringObject[]> ] [-ManagementPack] <ManagementPack> [-
Monitor] <ManagementPackMonitor[]> [[-Enforce]] [[-PassThru]] [-Confirm] [-WhatIf] [
<CommonParameters>]
```

Parameter Set: FromManagementPackClass

```
Enable-SCOMMonitor [-Class] <ManagementPackClass[]> [-ManagementPack] <ManagementPack> [-
Monitor] <ManagementPackMonitor[]> [[-Enforce]] [[-PassThru]] [-Confirm] [-WhatIf] [
<CommonParameters>]
```

### Detailed Description

The Enable-SCOMMonitor cmdlet enables monitors by creating and saving overrides for the specified monitors.

### Parameters

#### **-Class<ManagementPackClass[]>**

Specifies one or more class objects for which the monitors will be enabled. Enter a variable that represents the classes, or type a command that gets the classes. For information about how to get a class object, type Get-Help Get-SCOMClass.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	true (ByValue)

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

## **-Enforce**

Indicates that the enforce property will be set to True on the override that is created to enable the specified monitor.

Aliases	none
Required?	false
Position?	3
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-Group<MonitoringObject[]>**

Specifies one or more group objects for which the monitors will be enabled. Enter a variable that represents the groups, or type a command that gets the groups. For information about how to get a group object, type Get-Help Get-SCOMGroup.

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

## **-Instance<MonitoringObject[]>**

Specifies one or more class instance objects for which the monitors will be enabled. Enter a variable that represents the class instances, or type a command that gets the class instances. This parameter also accepts group objects. For information about how to get a class instance object, type Get-Help Get-SCOMClassInstance.

Aliases	none
Required?	false

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

### **-ManagementPack<ManagementPack>**

Specifies one or more management pack objects into which the enabling override will be saved. If the monitor is contained in an unsealed management pack, the overrides must be saved into the same management pack. Enter a variable that represents the management packs, or type a command that gets the management packs. For information about how to get a management pack object, type Get-Help Get-SCOMManagementPack.

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

### **-Monitor<ManagementPackMonitor[]>**

Specifies one or more monitor objects. Enter a variable that represents the monitors, or type a command that gets the monitors.

For information about how to get a monitor object, type Get-Help Get-SCOMMonitor.

Aliases	none
Required?	true
Position?	2
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-PassThru**

Returns an object representing the monitor. By default, this cmdlet does not generate any output.

Aliases	none
Required?	false
Position?	4
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-Confirm**

Prompts you for confirmation before executing the command.

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

## **-WhatIf**

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

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

## **<CommonParameters>**

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

# Examples

## ----- EXAMPLE 1 -----

Description

-----

The first three commands get a management pack object, a class object, and a monitor object, and then store the objects in the \$MP, \$Class, and \$Monitor variables, respectively.

The fourth command enables the monitor stored in \$Monitor for the management pack stored in \$MP using the Enforce property.

```
PS C:\>$MP = Get-SCOMManagementPack -displayname "My SQL MP Customization" | where  
{$_Sealed -eq $False}
```

```
PS C:\>$Class = Get-SCOMClass -DisplayName "SQL DB Engine"
```

```
PS C:\>$Monitor = Get-SCOMMonitor -DisplayName *memory*
```

```
PS C:\>Enable-SCOMMonitor -Class $Class -ManagementPack $MP -Monitor $Monitor -Enforce
```

## Related topics

[Get-SCOMClass](#)

[Get-SCOMClassInstance](#)

[Get-SCOMGroup](#)

[Get-SCOMManagementPack](#)

[Disable-SCOMMonitor](#)

[Get-SCOMMonitor](#)

# Enable-SCOMNotificationSubscription

---

## Enable-SCOMNotificationSubscription

Enables a notification subscription.

### Syntax

Parameter Set: Default

```
Enable-SCOMNotificationSubscription [-Subscription] <NotificationSubscription[]> [-PassThru] [ <CommonParameters>]
```

## Detailed Description

The Enable-SCOMNotificationSubscription cmdlet enables a notification subscription.

## Parameters

### -PassThru

Returns an object representing the notification subscription. By default, this cmdlet does not generate any output.

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

### -Subscription<NotificationSubscription[]>

Specifies which subscriptions to change.

Aliases	none
Required?	true
Position?	1

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

## <CommonParameters>

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

## Examples

### ----- EXAMPLE 1 -----

This command enables all disabled notifications.

```
PS C:\> Get-SCOMNotificationSubscription | where-object{$_ .Enabled -eq $false} | Enable-SCOMNotificationSubscription
```

### ----- EXAMPLE 2 -----

This command enables all notifications where the user CONTOSO\AIYoung is on the TO line.

```
PS C:\> Get-SCOMNotificationSubscription | where-object{$_ .ToRecipients -contains "CONTOSO\AIYoung"} | Enable-SCOMNotificationSubscription
```



# Enable-SCOMOperationalDataReporting

---

## Enable-SCOMOperationalDataReporting

Enables operational data reporting for the management group.

### Syntax

Parameter Set: Default

```
Enable-SCOMOperationalDataReporting [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

### Detailed Description

The Enable-SCOMOperationalDataReporting cmdlet enables operational data reporting for the management group.

### Parameters

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

Specifies a computer to establish a connection with. The computer must be running the System Center Data Access service. The default is the computer for the current management group connection.

Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

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

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

Specifies a user account under which the management group connection will run. It must have access to the server that is specified in the ComputerName parameter if specified. The default is the current user.

Enter a PSCredential object, such as one that is returned by the Get-Credential cmdlet.

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

## **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection.

Enter a Management group connection object, such as one returned by the Get-SCManagementGroupConnection cmdlet.

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

## **<CommonParameters>**

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

## **Examples**

### **----- EXAMPLE 1 -----**

This command enables operational data reporting.

```
PS C:\>Enable-SCOMOperationalDataReporting
```

# Enable-SCOMRule

---

## Enable-SCOMRule

Enables monitoring rules by creating and saving overrides for the specified rules.

### Syntax

Parameter Set: Empty

```
Enable-SCOMRule [-ManagementPack] <ManagementPack> [-Rule] <ManagementPackRule[]> [[-Enforce]] [[-PassThru]] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: FromGroup

```
Enable-SCOMRule [[-Group] <MonitoringObject[]> ] [-ManagementPack] <ManagementPack> [-Rule] <ManagementPackRule[]> [[-Enforce]] [[-PassThru]] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: FromInstance

```
Enable-SCOMRule [[-Instance] <MonitoringObject[]> ] [-ManagementPack] <ManagementPack> [-Rule] <ManagementPackRule[]> [[-Enforce]] [[-PassThru]] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: FromManagementPackClass

```
Enable-SCOMRule [-Class] <ManagementPackClass[]> [-ManagementPack] <ManagementPack> [-Rule] <ManagementPackRule[]> [[-Enforce]] [[-PassThru]] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The Enable-SCOMRule enables monitoring rules by creating and saving overrides for the specified rules.

### Parameters

#### **-Class<ManagementPackClass[]>**

Specifies one or more class objects for which the rules will be enabled. Enter a variable that represents the classes, or type a command that gets the classes. For information about how to get a class object, type Get-Help Get-SCOMClass.

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

## **-Enforce**

Indicates that the enforce property will be set to True on the override that is created to enable the specified monitoring rules.

Aliases	none
Required?	false
Position?	3
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-Group<MonitoringObject[]>**

Specifies one or more group objects for which the rules will be enabled. Enter a variable that represents the groups, or type a command that gets the groups. For information about how to get a group object, type Get-Help Get-SCOMGroup.

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

## **-Instance<MonitoringObject[]>**

Specifies one or more class instance objects for which the rules will be enabled. Enter a variable that represents the class instances, or type a command that gets the class instances. This parameter also accepts group objects. For information about how to get a class instance object, type Get-Help Get-SCOMClassInstance.

Aliases	none
Required?	false
Position?	1

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

### **-ManagementPack<ManagementPack>**

Specifies one or more management pack objects into which the enabling override will be saved. If the rule is contained in an unsealed management pack, the overrides must be saved into the same management pack. Enter a variable that represents the management packs, or type a command that gets the management packs. For information about how to get a management pack object, type Get-Help Get-SCOMManagementPack.

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

### **-PassThru**

Returns an object representing the monitoring rule. By default, this cmdlet does not generate any output.

Aliases	none
Required?	false
Position?	4
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-Rule<ManagementPackRule[]>**

Specifies one or more monitoring rule objects. Enter a variable that represents the monitoring rules, or type a command that gets the monitoring rules.

For information about how to get a monitoring rule object, type Get-Help Get-SCOMRule.

Aliases	none
Required?	true
Position?	2
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-Confirm**

Prompts you for confirmation before executing the command.

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

## **-WhatIf**

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

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

## **<CommonParameters>**

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

## Examples

### ----- EXAMPLE 1 -----

Description

-----

The first three commands get a management pack object, a class object, and a monitoring rule object, and then store the objects in the \$MP, \$Class, and \$Rule variables, respectively. The last command enables the rule \$Rule for the management pack \$MP using the Enforce property.

```
PS C:\>$MP = Get-SCOMManagementPack -displayname "My SQL MP Customization" | where  
{$_.Sealed -eq $False}
```

```
PS C:\>$Class = Get-SCOMClass -DisplayName "SQL DB Engine"
```

```
PS C:\>$Rule = Get-SCOMRule -DisplayName "*Events/sec"
```

```
PS C:\>Enable-SCOMRule -Class $Class -Rule $Rule -ManagementPack $MP -Enforce
```

## Related topics

[Get-SCOMClass](#)

[Get-SCOMClassInstance](#)

[Get-SCOMGroup](#)

[Get-SCOMManagementPack](#)

[Disable-SCOMRule](#)

[Get-SCOMRule](#)

# Exit-SCOMCEIP

---

## Exit-SCOMCEIP

Removes the local computer from SCOM CEIP data collection.

### Syntax

Parameter Set: Empty

```
Exit-SCOMCEIP [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The Exit-SCOMCEIP cmdlet removes the local computer from Operations Manager CEIP data collection.

### Parameters

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

Specifies a computer to establish a connection with. The computer must be running the System Center Data Access service. The default is the computer for the current management group connection.

Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

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

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

Specifies a user account under which the management group connection will run. It must have access to the server that is specified in the ComputerName parameter if specified. The default is the current user.

Enter a PSCredential object, such as one that is returned by the Get-Credential cmdlet.



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

### **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection.

Enter a Management group connection object, such as one returned by the Get-SCManagementGroupConnection cmdlet.

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

### **-Confirm**

Prompts you for confirmation before executing the command.

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

### **-WhatIf**

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

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

## <CommonParameters>

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

## Examples

### ----- EXAMPLE 1 -----

This command removes the computer from participating in Operations Manager CEIP.

```
PS C:\>Exit-SCOMCEIP
```

# Export-SCOMEffektivemonitoringConfiguration

---

## Export-SCOMEffektivemonitoringConfiguration

Retrieves the rules, monitors, and overrides that apply to a specified monitoring object, calculates the effective configuration of the rules and monitors, and then sends the output to a CSV file that can be imported into a program such as Microsoft Excel. The delineator used in the CSV file is the pipeline symbol, "|".

### Syntax

Parameter Set: Empty

```
Export-SCOMEffektivemonitoringConfiguration [-Instance] <MonitoringObject> [-Path] <String> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-Encoding <Encoding> ] [-RecurseContainedObjects] [-SCSession <Connection[]> ] [ <CommonParameters> ]
```

### Detailed Description

The Export-SCOMEffektivemonitoringConfiguration cmdlet retrieves the rules, monitors, and overrides that apply to a specified monitoring object, calculates the effective configuration of the rules and monitors, and then sends the output to a CSV file that can be imported into a program such as Microsoft Excel. The delineator used in the CSV file is the pipeline symbol, "|".

### Parameters

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

Specifies the name of a computer to establish a connection with. The computer must be running the System Center Data Access Service. The default value is the computer for the current management group connection. Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

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

## **-Credential<PSCredential>**

Specifies a user account under which the management group connection will run. The default is the current user. Type a user name, such as "User01", "Domain01\User01", or "User@Domain.com", or enter a PSCredential object, such as one returned by the Get-Credential cmdlet. When you type a user name, you will be prompted for a password. For more information, type Get-Help Get-Credential.

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

## **-Encoding<Encoding>**

Specifies the encoding to use for the monitoring configuration.

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

## **-Instance<MonitoringObject>**

Specifies one or more class instance objects. Enter a variable that represents the class instances, or type a command that gets the class instances. This parameter also accepts group objects. For information about how to get a class instance object, type Get-Help Get-SCOMClassInstance.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	true (ByValue)

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

### **-Path<String>**

Specifies the path and filename where the resulting CSV file should be stored. A file extension is not appended by default, therefore one must be provided.

Aliases	none
Required?	true
Position?	2
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-RecurseContainedObjects**

Indicates whether the data for objects that are contained by the MonitoringObject are included in the output. For example, in the case of a computer, RecurseContainedObjects returns all discoveries and monitors on the computer and all monitoring objects hosted on the computer, such as disks or network cards.

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

### **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection. Specify a Management group connection object, such as one returned by the Get-SCOMManagementGroupConnection cmdlet.

Aliases	none
Required?	false

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

## <CommonParameters>

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

## Examples

### ----- EXAMPLE 1 -----

Description

-----

The first command gets the group members of the All Windows Computers group and stores the objects in the \$Members object array. The second command uses the pipeline operator to pass each object stored in \$Members to a ForEach statement that exports the data to a CSV file at the specified location, including the name of the computer in the file name, and appending a .csv extension.

```
PS C:\> $members = (Get-SCOMGroup -DisplayName "All Windows  
Computers").GetRelatedMonitoringObjects()
```

```
PS C:\> $members | foreach { Export-SCOMEffctiveMonitoringConfiguration -Instance $_ -Path  
"C:\temp\${$_.DisplayName}.csv" } -RecurseContainedObjects }
```

# Export-SCOMManagementPack

---

## Export-SCOMManagementPack

Exports a management pack as an unsealed, valid XML-formatted file that you can later import into Operations Manager.

### Syntax

Parameter Set: FromManagementPack

```
Export-SCOMManagementPack [-ManagementPack] <ManagementPack[]> -Path <String> [-PassThru] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The Export-SCOMManagementPack cmdlet exports a management pack as an unsealed, valid XML-formatted file that you can later import into Operations Manager. All of the management pack's information is included in the file as XML data. You can use this cmdlet to save or archive management pack information.

### Parameters

#### **-ManagementPack<ManagementPack[]>**

Specifies one or more management packs to export. You can obtain a ManagementPack object by using the Get-SCOMManagementPack cmdlet.

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

#### **-PassThru**

Specifies the output object that represents the management pack. This output object can be passed to other cmdlets.

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

### **-Path<String>**

Specifies the folder into which exported management pack files will be stored. The specified folder must exist before you run the 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](#)

## Inputs

The input type is the type of the objects that you can pipe to the cmdlet.

- **Microsoft.EnterpriseManagement.Configuration.ManagementPack** You can pipe a management pack to the ManagementPack parameter of the Export-SCOMManagementPack cmdlet, for example, the object that is returned by the Get-SCOMManagementPack cmdlet.

## Outputs

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

- **Microsoft.EnterpriseManagement.Configuration.ManagementPack** When you use the PassThru parameter, the Export-SCOMManagementPack cmdlet returns a ManagementPack object.

## Examples

### ----- EXAMPLE 1 -----

The command in this example exports all management packs with a name that is similar to '\*snmp\*'.  
PS C:\>Get-SCOMManagementPack -Name snmp | Export-SCOMManagementPack -Path C:\temp;dir C:\temp\\*snmp\*

## Related topics

[Import-SCOMManagementPack](#)

[New-SCOMManagementPackBundle](#)

[Get-SCOMManagementPack](#)

[New-SCOMManagementPack](#)

[Protect-SCOMManagementPack](#)

[Remove-SCOMManagementPack](#)

[Test-SCOMManagementPack](#)

# Get-SCOMAccessLicense

---

## Get-SCOMAccessLicense

Gets information about licenses for System Center Operations Manager and Windows.

### Syntax

Parameter Set: Empty

```
Get-SCOMAccessLicense [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromLicense

```
Get-SCOMAccessLicense [-ShowLicense][ -ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromLicenseName

```
Get-SCOMAccessLicense [-LicenseName] <String[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

### Detailed Description

Gets a list of System Center Operation Manager license types or a list of computers and their license information. This cmdlet is intended to be used by a central management system as the license information needs to be correlated with and unduplicated against Access License data from other System Center products.

### Parameters

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

Specifies the name of a computer to establish a connection with. The computer must be running the System Center Data Access Service. The default value is the computer for the current management group connection. Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

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

## **-Credential<PSCredential>**

Specifies a user account under which the management group connection will run. The default is the current user. Type a user name, such as "User01", "Domain01\User01", or "User@Domain.com", or enter a PSCredential object, such as one returned by the Get-Credential cmdlet. When you type a user name, you will be prompted for a password. For more information, type Get-Help Get-Credential.

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

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

Specifies the license name for which instances will be retrieved. Valid values can be obtained by running the following: Get-ScomAccessLicense -License

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

## **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection. Specify a Management group connection object, such as one returned by the Get-SCOMManagementGroupConnection cmdlet.

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

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

## **-ShowLicense**

Displays a list of available licenses that apply to this product.

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

## **<CommonParameters>**

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

## **Examples**

### **----- EXAMPLE 1 -----**

Description

-----

This command shows the default experience for using this cmdlet.

```
PS C:\> Get-SCOMAccessLicense
```

### **----- EXAMPLE 2 -----**

Description

-----

This command shows how to get a list of available licenses that apply to the product.

```
PS C:\> Get-SCOMAccessLicense -License
```

### **----- EXAMPLE 3 -----**

Description

-----

This command shows how to get all license objects that have a LicenseName of 'System Center Operations Manager 2012 Management Server'.

```
PS C:\> Get-SCOMAccessLicense -LicenseName "System Center Operations Manager 2012 Management Server"
```

#### ----- **EXAMPLE 4** -----

Description

-----

This command shows how to get all license object that have a LicenseName that match 'SML'.

```
PS C:\> Get-SCOMAccessLicense -LicenseName "SML"
```

# Get-SCOMADAgentAssignment

---

## Get-SCOMADAgentAssignment

Gets active directory agent assignments for the management group.

### Syntax

Parameter Set: FromDomain

```
Get-SCOMADAgentAssignment [[-Domain] <String[]> ] [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromManagementServer

```
Get-SCOMADAgentAssignment [[-Domain] <String[]> ] -PrimaryServer <ManagementServer[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCOMADAgentAssignment cmdlet gets active directory agent assignments for the management group.

### Parameters

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

Specifies a computer to establish a connection with. The computer must be running the System Center Data Access service. The default is the computer for the current management group connection.

Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

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

## **-Credential<PSCredential>**

Specifies a user account under which the management group connection will run. It must have access to the server that is specified in the ComputerName parameter if specified. The default is the current user.

Enter a PSCredential object, such as one that is returned by the Get-Credential cmdlet.

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

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

Only AD agent assignments for this domain will be retrieved.

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

## **-PrimaryServer<ManagementServer[]>**

Only AD agent assignments to this primary management server will be retrieved.

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

## **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection.

Enter a Management group connection object, such as one returned by the Get-SCManagementGroupConnection cmdlet.

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

## **<CommonParameters>**

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

## **Examples**

### **----- EXAMPLE 1 -----**

This command gets all AD agent assignments for the domain01 domain

```
PS C:\>Get-SCOMADAgentAssignment -Domain 'domain.contoso.com'
```

### **----- EXAMPLE 2 -----**

This command gets all AD agent assignments with the primary server of OMServer01 and domain contoso.com

```
PS C:\>Get-SCOMManagementServer "OMServer01*" | Get-SCOMADAgentAssignment -Domain "contoso.com"
```



# Get-SCOMAgent

---

## Get-SCOMAgent

Gets the agent-managed computers in a management group.

### Syntax

Parameter Set: Empty

```
Get-SCOMAgent [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromAgentNames

```
Get-SCOMAgent [-DNSHostName] <String[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromManagementServer

```
Get-SCOMAgent [-ManagementServer] <ManagementServer> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCOMAgent cmdlet gets the agent-managed computers in a management group.

### Parameters

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

Specifies the name of a computer to establish a connection with. The computer must be running the System Center Data Access Service. The default value is the computer for the current management group connection. Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

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

## **-Credential<PSCredential>**

Specifies a user account under which the management group connection will run. The default is the current user. Type a user name, such as "User01", "Domain01\User01", or "User@Domain.com", or enter a PSCredential object, such as one returned by the Get-Credential cmdlet. When you type a user name, you will be prompted for a password. For more information, type Get-Help Get-Credential.

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

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

Specifies the name of a Domain Name System (DNS) host.

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

## **-ManagementServer<ManagementServer>**

Specifies the primary management server from which to retrieve all child agents. This parameter requires a management server object. For information about how to get a management server object, type Get-Help Get-SCOMManagementServer.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	true (ByValue, ByPropertyName)

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

## **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection. Specify a Management group connection object, such as one returned by the Get-SCOMManagementGroupConnection cmdlet.

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

## **<CommonParameters>**

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

## **Outputs**

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

- **Microsoft.EnterpriseManagement.Administration.AgentManagedComputer**

## **Examples**

### **----- EXAMPLE 1 -----**

Description

-----

This command establishes a temporary connection with Server01 and gets all agents in the management group.

```
PS C:\>Get-SCOMAgent -ComputerName "Server01.Contoso.Com"
```

### **----- EXAMPLE 2 -----**

Description

-----

The first command gets the management server named "MgmtServer01.Contoso.com" and stores it in the variable \$MgmtServer.

The second command gets the agents that are managed by the management server stored in the variable \$MgmtServer.

```
PS C:\>$MgmtServer = Get-SCOMManagementServer "MgmtServer01.Contoso.com"
```

```
PS C:\>Get-SCOMAgent -ManagementServer $MgmtServer
```

### ----- EXAMPLE 3 -----

Description

-----

This command gets agents that are named "Server01.Contoso.com", that begin with "Server02", and that are in the domain "Contoso.com".

```
PS C:\>Get-SCOMAgent -Name "Server01.Contoso.com", "Server02*", "*.Contoso.com"
```

### ----- EXAMPLE 4 -----

Description

-----

This command establishes a temporary connection with Server01.Contoso.com and gets all agents in the Contoso.com domain.

```
PS C:\>Get-SCOMAgent -Name "*.Contoso.com" -ComputerName "Server01.Contoso.com"
```

## Related topics

[Install-SCOMAgent](#)

[Uninstall-SCOMAgent](#)

# Get-SCOMAgentApprovalSetting

---

## Get-SCOMAgentApprovalSetting

Gets the manual agent approval setting for the management group.

### Syntax

Parameter Set: Empty

```
Get-SCOMAgentApprovalSetting [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCOMAgentApprovalSetting cmdlet gets the manual agent approval setting for the management group.

### Parameters

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

Specifies a computer to establish a connection with. The computer must be running the System Center Data Access service. The default is the computer for the current management group connection.

Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

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

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

Specifies a user account under which the management group connection will run. It must have access to the server that is specified in the ComputerName parameter if specified. The default is the current user.

Enter a PSCredential object, such as one that is returned by the Get-Credential cmdlet.

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

### **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection.

Enter a Management group connection object, such as one returned by the Get-SCManagementGroupConnection cmdlet.

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

### **<CommonParameters>**

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

### **Examples**

#### **----- EXAMPLE 1 -----**

This command gets the manual gent approval setting for the management group.

```
PS C:\>Get-SCOMAgentApprovalSetting
```

# Get-SCOMAgentlessManagedComputer

---

## Get-SCOMAgentlessManagedComputer

Retrieves a list managed computers that do not have agents.

### Syntax

Parameter Set: Empty

```
Get-SCOMAgentlessManagedComputer [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromAgentManagedBy

```
Get-SCOMAgentlessManagedComputer [-ManagedByAgent] <AgentManagedComputer[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromManagementServerManagedBy

```
Get-SCOMAgentlessManagedComputer [-ManagedByManagementServer] <ManagementServer[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromName

```
Get-SCOMAgentlessManagedComputer [-DNSHostName] <String[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCOMAgentlessManagedComputer cmdlet gets a list of managed computers that do not have agents.

### Parameters

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

Specifies the name of a computer to establish a connection with. The computer must be running the System Center Data Access Service. The default value is the computer for the current management group connection. Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

Aliases	none
Required?	false
Position?	named
Default Value	none

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

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

Specifies a user account under which the management group connection will run. The default is the current user. Type a user name, such as "User01", "Domain01\User01", or "User@Domain.com", or enter a PSCredential object, such as one returned by the Get-Credential cmdlet. When you type a user name, you will be prompted for a password. For more information, type Get-Help Get-Credential.

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

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

Specifies the name of a Domain Name System (DNS) host.

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

### **-ManagedByAgent<AgentManagedComputer[]>**

Specifies the agent that will perform the agentless management.

Aliases	none
Required?	true
Position?	1



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

### **-ManagedByManagementServer<ManagementServer[]>**

Specifies the management server that will perform the agentless management.

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

### **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection. Specify a Management group connection object, such as one returned by the Get-SCOMManagementGroupConnection cmdlet.

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

### **<CommonParameters>**

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

## Examples

### ----- EXAMPLE 1 -----

This command uses the Name parameter which can accept multiple strings that are full or have wildcards.

```
PS C:\> Get-SCOMAgentlesslyManagedComputer -Name server01,server0*
```

### ----- EXAMPLE 2 -----

This command gets a list of all of all agentless managed computers managed by the server \\server01.

```
PS C:\>Get-SCAgent server01 | foreach{Get-SCAgentlesslyManagedComputer -ManagedBy $_}Get-SCOMAgent server01 |foreach{ Get-SCOMAgentlesslyManagedComputer -ManagedByAgent $_ }
```

# Get-SCOMAlert

---

## Get-SCOMAlert

Gets the specified alerts.

### Syntax

Parameter Set: Empty

```
Get-SCOMAlert [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromAlertDefault

```
Get-SCOMAlert [[-Instance] <EnterpriseManagementObject[]> ] [[-LastModifiedBy] <String[]> ] [[-Name] <String[]> ] [[-Owner] <String[]> ] [[-ResolutionState] <Int32[]> ] [[-ResolvedBy] <String[]> ] [[-HealthState] <String[]> ] [[-Priority] <String[]> ] [[-Severity] <String[]> ] [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromAlertId

```
Get-SCOMAlert [-Id] <Guid[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromCriteria

```
Get-SCOMAlert [[-Criteria] <String> ] [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

## Detailed Description

The Get-SCOMAlert cmdlet gets the specified alerts.

## Parameters

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

Specifies the name of a computer to establish a connection with. The computer must be running the System Center Data Access Service. The default value is the computer for the current management group connection. Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

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

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

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

Specifies a user account under which the management group connection will run. The default is the current user. Type a user name, such as "User01", "Domain01\User01", or "User@Domain.com", or enter a PSCredential object, such as one returned by the Get-Credential cmdlet. When you type a user name, you will be prompted for a password. For more information, type Get-Help Get-Credential.

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

### **-Criteria<String>**

Aliases	none
Required?	false
Position?	10
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

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

Aliases	none
Required?	false
Position?	7
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-Id<Guid[]>**

Retrieves the alert with the specified GUID. The Id is stored in the Id property of the object that represents an alert. To get the Id of an alert, type "Get-SCOMAlert | Format-Table Name, Id".

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

### **-Instance<EnterpriseManagementObject[]>**

Retrieves alerts for one or more class instance objects. Enter a variable that represents the class instances, or type a command that gets the class instances. This parameter also accepts group objects. For information about how to get a class instance object, type Get-Help Get-SCOMClassInstance.

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

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

Retrieves alerts that match the specified user name for the last user that edited the alert.

Aliases	none
Required?	false
Position?	2
Default Value	none
Accept Pipeline Input?	true (ByValue)

Accept Wildcard Characters?	true
-----------------------------	------

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

Specifies the name of the alerts to retrieve.

Aliases	none
Required?	false
Position?	3
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	true

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

Retrieves alerts that match the specified owner for the alert.

Aliases	none
Required?	false
Position?	4
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	true

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

Aliases	none
Required?	false
Position?	8
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-ResolutionState<Int32[]>**

Retrieves alerts that match a specified resolution state Id. For example, the resolution state Id for "Closed" is 255.

Aliases	none
Required?	false
Position?	5
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	true

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

Retrieves alerts that match the specified user name for the user that resolved the alert.

Aliases	none
Required?	false
Position?	6
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	true

## **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection. Specify a Management group connection object, such as one returned by the Get-SCOMManagementGroupConnection cmdlet.

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

## -Severity<String[]>

Aliases	none
Required?	false
Position?	9
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## <CommonParameters>

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

## Examples

### ----- EXAMPLE 1 -----

Description

-----

This command gets all alerts with a resolution state of zero (new alerts).

```
PS C:\>Get-SCOMAlert -ResolutionSate 0
```

### ----- EXAMPLE 2 -----

Description

-----

This command gets all alerts that have a value of "TestServer" in their CustomField1 property.

```
PS C:\>Get-SCOMAlert | where {$_.CustomField1 -eq "TestServer"}
```

### ----- EXAMPLE 3 -----

Description

-----

The first command sets the variable \$OriginalErrorAction to the current value of the \$ErrorActionPreference variable.

The second command changes the value of \$ErrorActionPreference to "SilentlyContinue". "SilentlyContinue" will allow the following command to continue to run when it encounters a class instance that does not have alerts, and it will not display an error.



The third command gets all classes with "health" in their name and uses the pipeline operator (|) to pass the class objects to the Get-SCOMClassInstance cmdlet which gets the class instances for each class object. It then passes each of the class instance objects to the Get-SCOMAlert cmdlet which returns the alerts that have a resolution state between 5 and 200, inclusive, for each class instance.

The fourth command sets the value for \$ErrorActionPreference back to the value stored in the \$OriginalErrorAction variable.

```
PS C:\>$OriginalErrorAction = $ErrorActionPreference
```

```
PS C:\>$ErrorActionPreference = "SilentlyContinue"
```

```
PS C:\>Get-SCOMClass -Name *health* | Get-SCOMClassInstance | Get-SCOMAlert -ResolutionState (5..200)
```

```
PS C:\>$ErrorActionPreference = $OriginalErrorAction
```

## ----- EXAMPLE 4 -----

Description

-----

This command gets the alert with the Id of 7413b06b-a95b-4ae3-98f2-dac9ff76dabd.

```
PS C:\>Get-SCOMAlert -Id 7413b06b-a95b-4ae3-98f2-dac9ff76dabd
```

## Related topics

[Get-SCOMClassInstance](#)

[Set-SCOMAlert](#)

# Get-SCOMAlertHistory

---

## Get-SCOMAlertHistory

Gets history entries for the specified alerts.

### Syntax

Parameter Set: FromAlertDefault

```
Get-SCOMAlertHistory [-Alert] <MonitoringAlert[]> [ <CommonParameters>]
```

### Detailed Description

The Get-SCOMAlertHistory cmdlet gets history entries for the specified alerts.

### Parameters

#### **-Alert<MonitoringAlert[]>**

Retrieves the history for the specified alert objects. Enter a variable that represents the alerts, or type a command that gets the alerts. For information about how to get an alert object, type Get-Help Get-SCOMAlert.

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

#### **<CommonParameters>**

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

## Examples

### ----- EXAMPLE 1 -----

Description

-----

This command gets all alerts with "heartbeat" in their name and uses the pipeline operator (|) to pass the alerts to Get-SCOMAlertHistory which retrieves the history for each alert. Format-Table displays the values for the ResolutionState, Owner, ModifiedBy, and Comments properties for each alert.

```
PS C:\>Get-SCOMAlert -Name *heartbeat* | Get-SCOMAlertHistory | Format-Table,  
ResolutionState, Owner, ModifiedBy, Comments
```

## Related topics

[Get-SCOMAlert](#)

# Get-SCOMAlertResolutionSetting

---

## Get-SCOMAlertResolutionSetting

Gets the automatic alert resolution setting for the management group.

### Syntax

Parameter Set: Empty

```
Get-SCOMAlertResolutionSetting [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCOMAlertResolutionSetting cmdlet gets the automatic alert resolution setting for the management group.

### Parameters

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

Specifies a computer to establish a connection with. The computer must be running the System Center Data Access service. The default is the computer for the current management group connection.

Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

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

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

Specifies a user account under which the management group connection will run. It must have access to the server that is specified in the ComputerName parameter if specified. The default is the current user.

Enter a PSCredential object, such as one that is returned by the Get-Credential cmdlet.

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

### **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection.

Enter a Management group connection object, such as one returned by the Get-SCManagementGroupConnection cmdlet.

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

### **<CommonParameters>**

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

### **Examples**

#### **----- EXAMPLE 1 -----**

This command gets the automatic alert resolution for the management group.

```
PS C:\>Get-SCOMAlertResolutionSetting
```

# Get-SCOMAlertResolutionState

---

## Get-SCOMAlertResolutionState

Gets the alert resolution states defined in the management group.

### Syntax

Parameter Set: Empty

```
Get-SCOMAlertResolutionState [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromManagementState

```
Get-SCOMAlertResolutionState -ResolutionStateCode <Byte[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromName

```
Get-SCOMAlertResolutionState -Name <String[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCOMAlertResolutionState cmdlet gets the alert resolution states defined in the management group.

### Parameters

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

Specifies a computer to establish a connection with. The computer must be running the System Center Data Access service. The default is the computer for the current management group connection.

Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

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

## **-Credential<PSCredential>**

Specifies a user account under which the management group connection will run. It must have access to the server that is specified in the ComputerName parameter if specified. The default is the current user.

Enter a PSCredential object, such as one that is returned by the Get-Credential cmdlet.

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

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

The name of the alert resolution state.

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

## **-ResolutionStateCode<Byte[]>**

The code of the alert resolution state.

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

## **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection.

Enter a Management group connection object, such as one returned by the Get-SCManagementGroupConnection cmdlet.

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

## **<CommonParameters>**

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

## **Examples**

### **----- EXAMPLE 1 -----**

This command gets all resolution states.

```
PS C:\>Get-SCOMAlertResolutionState
```

### **----- EXAMPLE 2 -----**

This command gets information about the resolution state with code 42.

```
PS C:\>Get-SCOMAlertResolutionState -ResolutionStateCode 42
```



# Get-SCOMClass

---

## Get-SCOMClass

Retrieves a class.

### Syntax

Parameter Set: \_\_AllParameterSets

```
Get-SCOMClass [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromClassDisplayName

```
Get-SCOMClass [-DisplayName] <String[]> [ <CommonParameters>]
```

Parameter Set: FromClassGuids

```
Get-SCOMClass [-Id] <Guid[]> [ <CommonParameters>]
```

Parameter Set: FromClassName

```
Get-SCOMClass [-Name] <String[]> [ <CommonParameters>]
```

Parameter Set: FromEMO

```
Get-SCOMClass [-Instance] <EnterpriseManagementObject[]> [ <CommonParameters>]
```

Parameter Set: FromManagementPack

```
Get-SCOMClass [-ManagementPack] <ManagementPack[]> [ <CommonParameters>]
```

### Detailed Description

The Get-SCOMClass cmdlet retrieves a class defined by Operations Manager or an imported management pack.

### Parameters

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

Specifies a computer to establish a connection with. The computer must be running the System Center Data Access service. The default value is the computer for the current management group connection.

Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name (FQDN). To specify the local computer, type the computer name, "localhost", or a dot (.).

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

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

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

Specifies a user account under which the management group connection will run. The account must have access to the server that is specified in the ComputerName parameter, if the server is specified. The default value is the current user.

You can enter a PSCredential object that is returned by the Get-Credential cmdlet.

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

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

Specifies the display name of the class to retrieve.

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

### **-Id<Guid[]>**

Specifies the Id of the class to retrieve. This may be a Guid or a string that will be converted to a Guid.

Aliases	none
Required?	true
Position?	1
Default Value	none

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

### **-Instance<EnterpriseManagementObject[]>**

Specifies an instance of a class to retrieve.

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

### **-ManagementPack<ManagementPack[]>**

Specifies one or more management packs containing the classes to retrieve.

You can enter a ManagementPack object that is returned by the Get-SCOMManagementPack cmdlet.

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

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

Specifies the name of a class to retrieve.

Aliases	none
Required?	true
Position?	1
Default Value	none

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

## **-SCSession<Connection[]>**

Specifies a connection to a management server. The default value is the current management group connection.

You can enter a management group connection object that is returned by the Get-SCOMManagementGroupConnection cmdlet.

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

## **<CommonParameters>**

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

## **Inputs**

The input type is the type of the objects that you can pipe to the cmdlet.

- **Microsoft.EnterpriseManagement.Common.EnterpriseManagementObject** You can pipe an instance of a management pack to the Instance parameter of the Get-SCOMClass cmdlet. The Microsoft.EnterpriseManagement.Common.EnterpriseManagementObject object is one the properties of the output object of the Get-SCOMClassinstance cmdlet.
- **System.Guid** You can pipe a Guid to the Id parameter of the Get-SCOMClass cmdlets.
- **Microsoft.EnterpriseManagement.Configuration.ManagementPackManagementPack** You can pipe a management pack to the ManagementPack parameter of the Get-SCOMClass cmdlet. This management pack object should contain the class to retrieve.
- **System.String** You can pipe a name to the Name parameter of the Get-SCOMClass cmdlet.

## **Outputs**

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

- **Microsoft.EnterpriseManagement.Configuration.ManagementPackClass** This cmdlet generates a management pack object.

## Examples

### ----- EXAMPLE 1 -----

The command in this example retrieves all classes that end with 'user'.

```
PS C:\>Get-SCOMClass -name *user
```

### ----- EXAMPLE 2 -----

The command in this example retrieves the class in which DisplayName equals "User".

```
PS C:\>Get-SCOMClass -DisplayName "User"
```

# Get-SCOMClassInstance

---

## Get-SCOMClassInstance

Gets a list of class instances.

### Syntax

Parameter Set: Empty

```
Get-SCOMClassInstance [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromEMODisplayNameParameterSetName

```
Get-SCOMClassInstance [-DisplayName] <String[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromEMOIdParameterSetName

```
Get-SCOMClassInstance -Id <Guid[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromEMONameParameterSetName

```
Get-SCOMClassInstance -Name <String[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromGroup

```
Get-SCOMClassInstance [-Group] <EnterpriseManagementObject[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromManagementPackClass

```
Get-SCOMClassInstance [-Class] <ManagementPackClass[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCOMClassInstance cmdlet gets a list of class instances.

### Parameters

#### **-Class<ManagementPackClass[]>**

Retrieves the class instance for one or more class objects. Enter a variable that represents the classes, or type a command that gets the classes. For information about how to get a class object, type Get-Help Get-SCOMClass.

Aliases	none
Required?	true
Position?	1

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

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

Specifies the name of a computer to establish a connection with. The computer must be running the System Center Data Access Service. The default value is the computer for the current management group connection. Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

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

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

Specifies a user account under which the management group connection will run. The default is the current user. Type a user name, such as "User01", "Domain01\User01", or "User@Domain.com", or enter a PSCredential object, such as one returned by the Get-Credential cmdlet. When you type a user name, you will be prompted for a password. For more information, type Get-Help Get-Credential.

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

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

Specifies the display name of an object. This parameter takes one or more strings and will look for matches in the DisplayName property of the given objects that this cmdlet works with. DisplayName

values may vary depending on which localized management packs are imported into the management group and the locale of the user running Windows PowerShell.

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

### **-Group<EnterpriseManagementObject[]>**

Retrieves the class instance for one or more group objects. Enter a variable that represents the groups, or type a command that gets the groups. For information about how to get a group object, type Get-Help Get-SCOMGroup.

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

### **-Id<Guid[]>**

Retrieves the class instance with the specified GUID. The GUID is stored in the Id property of the object that represents an SCOMClassInstance. To get the GUID of a class, type "Get-SCOMClassInstance | Format-Table DisplayName, Id".

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



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

Specifies the name of an object. This parameter takes one or more strings and will look for matches in the Name property of the given objects that this cmdlet works with.

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

## **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection. Specify a Management group connection object, such as one returned by the Get-SCOMManagementGroupConnection cmdlet.

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

## **<CommonParameters>**

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

## **Examples**

### **----- EXAMPLE 1 -----**

Description

-----

This command gets class instances that have "Server01.Contoso.Com" in their display name and class instances that are in the Contoso.com domain.

```
PS C:\>Get-SCOMClassInstance -Displayname "Server01.Contoso.Com", "*.Contoso.com"
```

## ----- EXAMPLE 2 -----

Description

-----

This command gets all classes with "Windows" in their name. It then uses the pipeline operator (|) to send the classes to the Get-SCOMClassInstance cmdlet which retrieves the class instances for each of the classes.

```
PS C:\>Get-SCOMClass -Name *Windows* | Get-SCOMClassInstance
```

## ----- EXAMPLE 3 -----

Description

-----

This command gets all groups with "Windows" in their name. It then uses the pipeline operator (|) to send the groups to the Get-SCOMClassInstance cmdlet which retrieves the class instances for each of the groups.

```
PS C:\>Get-SCOMGroup -DisplayName *Windows* | Get-SCOMClassInstance
```

## ----- EXAMPLE 4 -----

Description

-----

This command gets the class instance that has an Id of 7413b06b-a95b-4ae3-98f2-dac9ff76dabd.

```
PS C:\>Get-SCOMClassInstance -Id 7413b06b-a95b-4ae3-98f2-dac9ff76dabd
```

## Related topics

[Get-SCOMClass](#)

[Get-SCOMGroup](#)

# Get-SCOMCommand

---

## Get-SCOMCommand

Gets Operations Manager commands in the current session.

### Syntax

Parameter Set: CmdletSet

```
Get-SCOMCommand [-Noun <String[]> ] [-Verb <String[]> ] [ <CommonParameters>]
```

Parameter Set: AllCommandSet

```
Get-SCOMCommand [[-Name] <String[]> ] [-CommandType <CommandTypes> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCOMCommand cmdlet gets Operations Manager commands in the current session.

### Parameters

#### **-CommandType<CommandTypes>**

Specifies a connection to a management server. The default is the current management group connection.

Enter a Management group connection object, such as one returned by the Get-SCManagementGroupConnection cmdlet.

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

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

Specifies a connection to a management server. The default is the current management group connection.

Enter a Management group connection object, such as one returned by the Get-SCManagementGroupConnection cmdlet.

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

**-Noun<String[]>**

Specifies a connection to a management server. The default is the current management group connection.

Enter a Management group connection object, such as one returned by the Get-SCManagementGroupConnection cmdlet.

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

**-Verb<String[]>**

Specifies a connection to a management server. The default is the current management group connection.

Enter a Management group connection object, such as one returned by the Get-SCManagementGroupConnection cmdlet.

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 -----

This command gets all commands in the Operations Manager module with the word "agent" in their name.

```
PS C:\>Get-SCOMCommand "*agent*"
```

### ----- EXAMPLE 2 -----

This command gets all commands in the Operations Manager module that use the verb "Get".

```
PS C:\>Get-SCOMCommand -Verb Get
```

# Get-SCOMConnector

---

## Get-SCOMConnector

Gets the specified connectors.

### Syntax

Parameter Set: Empty

```
Get-SCOMConnector [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromConnectorDisplayName

```
Get-SCOMConnector [[-DisplayName] <String[]> ] [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromConnectorId

```
Get-SCOMConnector [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-Id <Guid[]> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromConnectorName

```
Get-SCOMConnector [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-Name <String[]> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCOMConnector cmdlet gets the specified connectors.

### Parameters

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

Specifies the name of a computer to establish a connection with. The computer must be running the System Center Data Access Service. The default value is the computer for the current management group connection. Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

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

## **-Credential<PSCredential>**

Specifies a user account under which the management group connection will run. The default is the current user. Type a user name, such as "User01", "Domain01\User01", or "User@Domain.com", or enter a PSCredential object, such as one returned by the Get-Credential cmdlet. When you type a user name, you will be prompted for a password. For more information, type Get-Help Get-Credential.

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

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

Specifies the display name of an object. This parameter takes one or more strings and will look for matches in the DisplayName property of the given objects that this cmdlet works with. DisplayName values may vary depending on which localized management packs are imported into the management group and the locale of the user running Windows PowerShell.

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

## **-Id<Guid[]>**

Retrieves the connector with the specified GUID. The Id is stored in the Id property of the object that represents an SCOM connector. To get the Id of a connector, type "Get-SCOMConnector | Format-Table Name, Id".

Aliases	none
Required?	false
Position?	named

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

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

Specifies the name of an object.

This parameter takes one or more strings and will look for matches in the Name property of the given objects that this cmdlet works with.

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

### **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection. Specify a Management group connection object, such as one returned by the Get-SCOMManagementGroupConnection cmdlet.

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

### **<CommonParameters>**

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



## Examples

### ----- EXAMPLE 1 -----

Description

-----

This command gets all connectors that have a display name beginning with "Connector".

```
PS C:\>Get-SCOMConnector -DisplayName Connector*
```

### ----- EXAMPLE 2 -----

Description

-----

This command gets all connectors that have a name that begins with "Operations Manager".

```
PS C:\>Get-SCOMConnector -Name "Operations Manager*"
```

### ----- EXAMPLE 3 -----

Description

-----

This command gets the connector with the Id of 7413b06b-a95b-4ae3-98f2-dac9ff76dabd.

```
PS C:\>Get-SCOMConnector -Id 7413b06b-a95b-4ae3-98f2-dac9ff76dabd
```

# Get-SCOMDatabaseGroomingSetting

---

## Get-SCOMDatabaseGroomingSetting

Gets the database grooming settings for the management group.

### Syntax

Parameter Set: Empty

```
Get-SCOMDatabaseGroomingSetting [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCOMDatabaseGroomingSetting cmdlet gets the database grooming settings for the management group. Database grooming settings are used to automatically remove unnecessary data from the Operations Manager database in order to maintain performance.

### Parameters

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

Specifies a computer to establish a connection with. The computer must be running the System Center Data Access service. The default is the computer for the current management group connection.

Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

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

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

Specifies a user account under which the management group connection will run. It must have access to the server that is specified in the ComputerName parameter if specified. The default is the current user.

Enter a PSCredential object, such as one that is returned by the Get-Credential cmdlet.

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

## **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection.

Enter a Management group connection object, such as one returned by the Get-SCManagementGroupConnection cmdlet.

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

## **<CommonParameters>**

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

## **Examples**

### **----- EXAMPLE 1 -----**

This example gets the database grooming settings for the management group.

```
PS C:\>Get-SCOMDatabaseGroomingSetting
```

# Get-SCOMDataWarehouseSetting

---

## Get-SCOMDataWarehouseSetting

Gets the data warehouse settings for the management group.

### Syntax

Parameter Set: Empty

```
Get-SCOMDataWarehouseSetting [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCOMDataWarehouseSetting cmdlet gets the data warehouse settings for the management group.

### Parameters

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

Specifies a computer to establish a connection with. The computer must be running the System Center Data Access service. The default is the computer for the current management group connection.

Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

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

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

Specifies a user account under which the management group connection will run. It must have access to the server that is specified in the ComputerName parameter if specified. The default is the current user.

Enter a PSCredential object, such as one that is returned by the Get-Credential cmdlet.

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

### **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection.

Enter a Management group connection object, such as one returned by the Get-SCManagementGroupConnection cmdlet.

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

### **<CommonParameters>**

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

### **Examples**

#### **----- EXAMPLE 1 -----**

This command gets the data warehouse settings for the management group.

```
PS C:\>Get-SCOMDataWarehouseSetting
```

# Get-SCOMDiagnostic

---

## Get-SCOMDiagnostic

Gets a list of diagnostics.

### Syntax

Parameter Set: Empty

```
Get-SCOMDiagnostic [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromDiagnosticDisplayName

```
Get-SCOMDiagnostic [-DisplayName] <String[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromDiagnosticId

```
Get-SCOMDiagnostic -Id <Guid[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromDiagnosticName

```
Get-SCOMDiagnostic -Name <String[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromManagementPack

```
Get-SCOMDiagnostic -ManagementPack <ManagementPack[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromManagementPackClass

```
Get-SCOMDiagnostic -Target <ManagementPackClass[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromMonitorInstance

```
Get-SCOMDiagnostic -Monitor <ManagementPackMonitor[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

## Detailed Description

The Get-SCOMDiagnostic cmdlet gets a list of diagnostics.

## Parameters

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

Specifies a computer to establish a connection with. The computer must be running the Operations Manager Data Access service.

Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

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

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

Specifies a user account under which the management group connection will run. The default is the current user.

Type a user name, such as "User01", "Domain01\User01", or "User@Domain.com", or enter a PSCredential object, such as one returned by the Get-Credential cmdlet. When you type a user name, you will be prompted for a password. For more information, type Get-Help Get-Credential.

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

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

Specifies the DisplayName. This parameter takes one or more strings and will look for matches in the DisplayName property of the given objects that this cmdlet works with.

The DisplayName values may vary based which localized management packs are imported into the management group as well as the locale of the user running PowerShell.

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

## **-Id<Guid[]>**

Specifies the globally unique identifier. This parameter takes one or more globally unique identifiers (Guids) and uses them to compare against the id property of the objects retrieved by this cmdlet.

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

## **-ManagementPack<ManagementPack[]>**

Specifies the management pack object. This parameter takes one or more management pack objects. Retrieve management pack objects using the Get-SCManagementPack cmdlet.

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

## **-Monitor<ManagementPackMonitor[]>**

Specifies the monitors that this cmdlet should retrieve objects for.

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



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

Specifies the Name. This parameter takes one or more strings and will look for matches in the Name property of the given objects that this cmdlet works with.

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

## **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection.

Specify a Management group connection object, such as one returned by the Get-SCManagementGroupConnection cmdlet.

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

## **-Target<ManagementPackClass[]>**

Specifies the target classes to use to limit the results of this cmdlet. Retrieve classes using the Get-SCClass cmdlet.

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

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

## <CommonParameters>

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

## Examples

### ----- EXAMPLE 1 -----

This command gets a list of all diagnostics related to Network Management.

```
PS C:\> Get-SCOMDiagnostic -Name *NetworkManagement*
```

# Get-SCOMDiscovery

---

## Get-SCOMDiscovery

Retrieves discovery information from Operations Manager.

### Syntax

Parameter Set: \_\_AllParameterSets

```
Get-SCOMDiscovery [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromDiscoveryDisplayName

```
Get-SCOMDiscovery [-DisplayName] <String[]> [ <CommonParameters>]
```

Parameter Set: FromDiscoveryId

```
Get-SCOMDiscovery [-Id] <Guid[]> [ <CommonParameters>]
```

Parameter Set: FromDiscoveryName

```
Get-SCOMDiscovery [-Name] <String[]> [ <CommonParameters>]
```

Parameter Set: FromManagementPack

```
Get-SCOMDiscovery [-ManagementPack] <ManagementPack[]> [ <CommonParameters>]
```

Parameter Set: FromManagementPackClass

```
Get-SCOMDiscovery [-Target] <ManagementPackClass[]> [ <CommonParameters>]
```

## Detailed Description

The Get-SCOMDiscovery cmdlet retrieves discovery information from Operations Manager.

## Parameters

### -ComputerName<String[]>

Specifies a computer to establish a connection with. The computer must be running the System Center Data Access service. The default value is the computer for the current management group connection.

Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name (FQDN). To specify the local computer, type the computer name, "localhost", or a dot (.).

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

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

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

Specifies a user account under which the management group connection will run. The account must have access to the server that is specified in the ComputerName parameter, if the server is specified. The default value is the current user.

You can enter a PSCredential object that is returned by the Get-Credential cmdlet.

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

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

Specifies the display name of the discovery object to be retrieved.

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

### **-Id<Guid[]>**

Specifies the Id (guid) of the discovery object to be retrieved. This may be a Guid or a string that will be converted to a Guid.

Aliases	none
Required?	true
Position?	1

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

### **-ManagementPack<ManagementPack[]>**

Specifies one or more management packs containing the discovery objects to retrieve.

You can enter a ManagementPack object that is returned by the Get-SCOMManagementPack cmdlet.

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

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

Specifies the name of a discovery object to retrieve.

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

### **-SCSession<Connection[]>**

Specifies a connection to a management server. The default value is the current management group connection.

You can enter a management group connection object that is returned by the Get-SCOMManagementGroupConnection cmdlet.

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

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

## **-Target<ManagementPackClass[]>**

Specifies the management pack that contains the classes that are targeted by the discovery.

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

## **<CommonParameters>**

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

## **Inputs**

The input type is the type of the objects that you can pipe to the cmdlet.

- **Microsoft.EnterpriseManagement.Configuration.ManagementPack**You can pipe a management pack object to the ManagementPack parameter of the Get-SCOMDiscovery cmdlet.
- **System.String**You can pipe a discovery name to the Name parameter of the Get-SCOMDiscovery cmdlet.
- **System.Guid**You can pipe a Guid of a discovery object to the Id parameter of the Get-SCOMDiscovery cmdlet.
- **Microsoft.EnterpriseManagement.Configuration.ManagementPackClass**You can pipe a management pack to the Target parameter of the Get-SCOMDiscovery cmdlet.

## **Outputs**

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

- **Microsoft.EnterpriseManagement.Configuration.ManagementPackDiscovery**This cmdlets generates a discovery object.

## Examples

### ----- EXAMPLE 1 -----

The command in this example retrieves the discoveries that are defined in Operations Manager.

```
PS C:\>Get-SCOMDiscovery
```

### ----- EXAMPLE 2 -----

Gets the discovery that is named '\*PopulateRootM\*' and formats the output as a list.

```
PS C:\>Get-SCOMDiscovery -Name *PopulateRootM*|format-list
```

# Get-SCOMErrorReportingSetting

---

## Get-SCOMErrorReportingSetting

Gets the error reporting settings for the management group.

### Syntax

Parameter Set: Empty

```
Get-SCOMErrorReportingSetting [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCOMErrorReportingSetting cmdlet gets the error reporting settings for the management group.

### Parameters

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

Specifies a computer to establish a connection with. The computer must be running the System Center Data Access service. The default is the computer for the current management group connection.

Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

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

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

Specifies a user account under which the management group connection will run. It must have access to the server that is specified in the ComputerName parameter if specified. The default is the current user.

Enter a PSCredential object, such as one that is returned by the Get-Credential cmdlet.



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

### **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection.

Enter a Management group connection object, such as one returned by the Get-SCManagementGroupConnection cmdlet.

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

### **<CommonParameters>**

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

### **Examples**

#### **----- EXAMPLE 1 -----**

This command gets the error reporting settings for the management group.

```
PS C:\>Get-SCOMErrorReportingSetting
```

# Get-SCOMEvent

---

## Get-SCOMEvent

Gets events that have been collected by monitoring rules.

### Syntax

Parameter Set: Empty

```
Get-SCOMEvent [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromEventId

```
Get-SCOMEvent [-Id] <Guid[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromInstance

```
Get-SCOMEvent [-Instance] <EnterpriseManagementObject[]> [[-EventId] <Int32[]> ] [[-EventLogName] <String[]> ] [[-EventSource] <String[]> ] [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromManagementPackRule

```
Get-SCOMEvent [-Rule] <ManagementPackRule[]> [[-EventId] <Int32[]> ] [[-EventLogName] <String[]> ] [[-EventSource] <String[]> ] [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCOMEvent cmdlet gets events that have been collected by monitoring rules.

### Parameters

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

Specifies the name of a computer to establish a connection with. The computer must be running the System Center Data Access Service. The default value is the computer for the current management group connection. Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

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

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

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

Specifies a user account under which the management group connection will run. The default is the current user. Type a user name, such as "User01", "Domain01\User01", or "User@Domain.com", or enter a PSCredential object, such as one returned by the Get-Credential cmdlet. When you type a user name, you will be prompted for a password. For more information, type Get-Help Get-Credential.

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

### **-EventId<Int32[]>**

Retrieves an event matching the specified ID. Event IDs are stored in the Number property of the object that represents an SCOM event. To get the event ID of an event, type "Get-SCOMEvent -Id \$Id | Format-Table RuleDisplayName, Number".

Aliases	none
Required?	false
Position?	2
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	true

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

Retrieves events matching the specified event log name. Event log names are stored in the Channel property of the object that represents an SCOM event. To get the event source of an event, type "Get-SCOMEvent -Id \$Id | Format-Table RuleDisplayName, Channel".

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

Required?	false
Position?	3
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	true

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

Retrieves events matching the specified event source. Event sources are stored in the PublisherName property of the object that represents an SCOMEvent. To get the event ID of an event, type "Get-SCOMEvent -Instance \$Instance | Format-Table DisplayName, PublisherName".

Aliases	none
Required?	false
Position?	4
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	true

### **-Id<Guid[]>**

Retrieves the event with the specified GUID.

The GUID is stored in the Id property of the object that represents an SCOMEvent. To get the GUID of an event, type "Get-SCOMEvent -Instance \$Instance | Format-Table RuleDisplayName, Id".

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

### **-Instance<EnterpriseManagementObject[]>**

Retrieves all events associated with the specified class instance object. Enter a variable that represents the class instance, or type a command that gets the class instance. This parameter also accepts group

objects. For information about how to get a class instance object, type Get-Help Get-SCOMClassInstance.

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

### **-Rule<ManagementPackRule[]>**

Retrieves all events collected by the specified monitoring rule object. Enter a variable that represents the monitoring rule, or type a command that gets the monitoring rule. For information about how to get a monitoring rule object, type Get-Help Get-SCOMRule.

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

### **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection. Specify a Management group connection object, such as one returned by the Get-SCOMManagementGroupConnection cmdlet.

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

## <CommonParameters>

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

## Examples

### ----- EXAMPLE 1 -----

Description

-----

The first command gets all monitoring rules with "WMI" in their name and stores them in the \$Rules variable.

The second command gets all classes with "health" in their display name and uses the pipeline operator (|) to pass the class objects to the Get-SCOMClassInstance cmdlet which gets the class instances for each of the class objects. Then, the command gets the events for each class instance matching the rules stored in the \$Rules variable. Using the ErrorAction parameter with its value set to SilentlyContinue allows the command to continue if it does not find an event that matches the specified rule.

```
PS C:\>$Rules = Get-SCOMRule -Name *WMI*
```

```
PS C:\>Get-SCOMClass -DisplayName *health* | Get-SCOMClassInstance | Get-SCOMEvent -Rule $Rules -ErrorAction SilentlyContinue
```

### ----- EXAMPLE 2 -----

Description

-----

This command gets the event with an Id of 7413b06b-a95b-4ae3-98f2-dac9ff76dabd.

```
PS C:\>Get-SCOMEvent -Id 7413b06b-a95b-4ae3-98f2-dac9ff76dabd
```

### ----- EXAMPLE 3 -----

Description

-----

This command gets all events and then pipes them to the Group-Object cmdlet which displays the events grouped by EventId.

```
PS C:\>Get-SCOMEvent | Group-Object -Property Number
```

## Related topics

[Get-SCOMClassInstance](#)

[Get-SCOMRule](#)

# Get-SCOMGatewayManagementServer

---

## Get-SCOMGatewayManagementServer

Gets the gateway management servers.

### Syntax

Parameter Set: Empty

```
Get-SCOMGatewayManagementServer [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromGatewayManagementServerNames

```
Get-SCOMGatewayManagementServer [[-Name] <String[]> ] [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCOMGatewayManagementServer cmdlet gets the gateway management servers.

### Parameters

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

Specifies the name of a computer to establish a connection with. The computer must be running the System Center Data Access Service. The default value is the computer for the current management group connection. Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

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

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

Specifies a user account under which the management group connection will run. The default is the current user. Type a user name, such as "User01", "Domain01\User01", or "User@Domain.com", or

enter a PSCredential object, such as one returned by the Get-Credential cmdlet. When you type a user name, you will be prompted for a password. For more information, type Get-Help Get-Credential.

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

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

Specifies the name of an object.

This parameter takes one or more strings and will look for matches in the Name property of the given objects that this cmdlet works with.

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

### **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection. Specify a Management group connection object, such as one returned by the Get-SCOMManagementGroupConnection cmdlet.

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



## <CommonParameters>

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

## Examples

### ----- EXAMPLE 1 -----

Description

-----

This command gets the gateway management server named "Server01.Contoso.com" and all gateway management servers that are in the Contoso.com domain.

```
PS C:\>Get-SCOMGatewayManagementServer -Name "Server01.Contoso.com", "*.Contoso.com"
```

### ----- EXAMPLE 2 -----

Description

-----

This command gets all gateway management servers that have a health state of "Success".

```
PS C:\>Get-SCOMGatewayManagementServer | where {$_.HealthState -eq "Success"}
```

### ----- EXAMPLE 3 -----

Description

-----

This command establishes a temporary connection to Server01.Contoso.com and gets all gateway management servers for that connection.

```
PS C:\>Get-SCOMGatewayManagementServer -ComputerName "Server01.Contoso.com"
```

## Related topics

[Get-SCOMManagementServer](#)

# Get-SCOMGroup

---

## Get-SCOMGroup

Gets a list of groups.

### Syntax

Parameter Set: Empty

```
Get-SCOMGroup [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromGroupName

```
Get-SCOMGroup [-DisplayName] <String[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromGroupGuid

```
Get-SCOMGroup [-Id] <Guid[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCOMGroup cmdlet gets a list of groups.

Because a group object is a type of class instance object, it can be passed into an Instance parameter.

### Parameters

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

Specifies the name of a computer to establish a connection with. The computer must be running the System Center Data Access Service. The default value is the computer for the current management group connection. Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

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

## **-Credential<PSCredential>**

Specifies a user account under which the management group connection will run. The default is the current user. Type a user name, such as "User01", "Domain01\User01", or "User@Domain.com", or enter a PSCredential object, such as one returned by the Get-Credential cmdlet. When you type a user name, you will be prompted for a password. For more information, type Get-Help Get-Credential.

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

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

Specifies the display name of an object. This parameter takes one or more strings and will look for matches in the DisplayName property of the given objects that this cmdlet works with. DisplayName values may vary depending on which localized management packs are imported into the management group and the locale of the user running Windows PowerShell.

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

## **-Id<Guid[]>**

Retrieves the group with the specified GUID.

The GUID is stored in the Id property of the object that represents an SCOMGroup. To get the GUID of a class, type "Get-SCOMGroup | Format-Table DisplayName, Id".

Aliases	none
Required?	true
Position?	1

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

## **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection. Specify a Management group connection object, such as one returned by the Get-SCOMManagementGroupConnection cmdlet.

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

## **<CommonParameters>**

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

## **Examples**

### **----- EXAMPLE 1 -----**

Description

-----

This command gets all groups that have a display name that includes "Agent" and all groups that have a display name that includes "Windows".

```
PS C:\>Get-SCOMGroup -DisplayName *Agent*, *Windows*
```

### **----- EXAMPLE 2 -----**

Description

-----

This command gets all groups that have "computer" in their display name.

```
PS C:\>Get-SCOMGroup *computer*
```

### ----- EXAMPLE 3 -----

Description

-----

This command gets the group with the Id of 7413b06b-a95b-4ae3-98f2-dac9ff76dabd.

```
PS C:\>Get-SCOMGroup -Id 7413b06b-a95b-4ae3-98f2-dac9ff76dabd
```

### **Related topics**

[Get-SCOMClassInstance](#)

# Get-SCOMHeartbeatSetting

---

## Get-SCOMHeartbeatSetting

Gets the server and agent heartbeat settings for the management group.

### Syntax

Parameter Set: Empty

```
Get-SCOMHeartbeatSetting [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCOMHeartbeatSetting cmdlet gets the server and agent heartbeat settings for the management group.

### Parameters

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

Specifies a computer to establish a connection with. The computer must be running the System Center Data Access service. The default is the computer for the current management group connection.

Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

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

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

Specifies a user account under which the management group connection will run. It must have access to the server that is specified in the ComputerName parameter if specified. The default is the current user.

Enter a PSCredential object, such as one that is returned by the Get-Credential cmdlet.

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

## **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection.

Enter a Management group connection object, such as one returned by the Get-SCManagementGroupConnection cmdlet.

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

## **<CommonParameters>**

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

## **Examples**

### **----- EXAMPLE 1 -----**

This command gets the server and agent heartbeat settings for the management group.

```
PS C:\>Get-SCOMHeartbeatSetting
```

# Get-SCOMLicense

---

## Get-SCOMLicense

Displays the Microsoft Software License Terms for the currently active product license.

### Syntax

Parameter Set: Empty

```
Get-SCOMLicense [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCOMLicense cmdlet displays the Microsoft Software License Terms for the currently active product license.

### Parameters

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

Specifies a computer to establish a connection with. The computer must be running the System Center Data Access service. The default is the computer for the current management group connection.

Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

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

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

Specifies a user account under which the management group connection will run. It must have access to the server that is specified in the ComputerName parameter if specified. The default is the current user.

Enter a PSCredential object, such as one that is returned by the Get-Credential cmdlet.



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

### **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection.

Enter a Management group connection object, such as one returned by the Get-SCManagementGroupConnection cmdlet.

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

### **<CommonParameters>**

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

### **Examples**

#### **----- EXAMPLE 1 -----**

This command displays the Microsoft Software License Terms for the currently active product license.

```
PS C:\>Get-SCOMLicense
```

# Get-SCOMLocation

---

## Get-SCOMLocation

Gets a location.

### Syntax

Parameter Set: Empty

```
Get-SCOMLocation [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromAgent

```
Get-SCOMLocation [-Agent] <AgentManagedComputer[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromDisplayName

```
Get-SCOMLocation [-DisplayName] <String[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromEMOIdParameterSetName

```
Get-SCOMLocation -Id <Guid[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromManagementServer

```
Get-SCOMLocation [-ManagementServer] <ManagementServer[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromPool

```
Get-SCOMLocation [-Pool] <ManagementServicePool[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCOMLocation cmdlet gets a location. You can associate an agent with a location so that its state can be displayed in the Web Application Availability Monitoring Summary Map Dashboard.

For more information about locations, type: Get-Help New-SCOMLocation -detailed.

### Parameters

#### **-Agent<AgentManagedComputer[]>**

Specifies one or more agent objects. Enter a variable that represents the agents, or type a command that gets the agents.

For information about how to get agents, type Get-Help Get-SCOMAgent.

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

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

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

Specifies a computer to establish a connection with. The computer must be running the Operations Manager Data Access service.

Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

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

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

Specifies a user account under which the management group connection will run. The default is the current user. Type a user name, such as "User01", "Domain01\User01", or "User@Domain.com", or enter a PSCredential object, such as one returned by the Get-Credential cmdlet. When you type a user name, you will be prompted for a password. For more information, type Get-Help Get-Credential.

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

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

Specifies the display name of an object. This parameter takes one or more strings and will look for matches in the DisplayName property of the given objects that this cmdlet works with. DisplayName values may vary depending on which localized management packs are imported into the management group and the locale of the user running Windows PowerShell.

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

## **-Id<Guid[]>**

Specifies the GUID of the location to retrieve.

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

## **-ManagementServer<ManagementServer[]>**

Specifies a primary management server object. For information about how to get a management server object, type Get-Help Get-SCOMManagementServer.

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

## **-Pool<ManagementServicePool[]>**

Specifies one or more resource pool objects. Enter a variable that represents the resource pools, or type a command that gets the resource pools.

For information about how to get a resource pool object, type "Get-Help Get-SCOMResourcePool".

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

## **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection.

For information about how to get a management group connection, type Get-Help Get-SCOMManagementGroupConnection cmdlet.

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

## **<CommonParameters>**

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

## **Examples**

### **----- EXAMPLE 1 -----**

Description

-----

This command gets the location with the display name "Seattle, WA" and displays information about this location to the user.

```
PS C:\> Get-SCOMLocation -DisplayName "Seattle, WA"
```

## ----- EXAMPLE 2 -----

Description

-----

The first command gets the management group connection object for the management group MgmtGroup01 and stores the object in the \$Session variable.

The second command gets the location for the session stored in \$Session.

```
PS C:\> $Session = Get-SCOMManagementGroupConnection -ManagementGroupName "MgmtGroup01"
PS C:\> Get-SCOMLocation -SCSession $Session
```

## ----- EXAMPLE 3 -----

Description

-----

The first command gets the agent-managed computer object named Server01 and stores the object in the \$Agent variable.

The second command gets the location for the agent-managed computer stored in \$Agent.

```
PS C:\> $Agent = Get-SCOMAgent -Name "Server01.Contoso.com"
PS C:\> Get-SCOMLocation -Agent $Agent
```

## ----- EXAMPLE 4 -----

Description

-----

The first command gets the management server object named MgmtServer01 and stores the object in the \$MgmtSvr variable.

The second command gets the location for the management server stored in \$MgmtSvr.

```
PS C:\> $MgmtSvr = Get-SCOMManagementServer -Name "MgmtServer01.Contoso.com"
PS C:\> Get-SCOMLocation -ManagementServer $MgmtSvr
```

## ----- EXAMPLE 5 -----

Description

-----

The first command gets the resource pool object named Pool01 and stores the object in the \$Pool variable.

The second command gets the location for the pool stored in \$Pool.

```
PS C:\> $Pool = Get-SCOMResourcePool -Name "Pool01"
```

```
PS C:\>Get-SCOMLocation -Pool $Pool
```

## Related topics

[New-SCOMLocation](#)

[Remove-SCOMLocation](#)

[Set-SCOMLocation](#)

# Get-SCOMMaintenanceMode

---

## Get-SCOMMaintenanceMode

Gets details about maintenance mode entries.

### Syntax

Parameter Set: FromInstance

```
Get-SCOMMaintenanceMode [[-Instance] <MonitoringObject[]> ] [[-History]] [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [  
<CommonParameters>]
```

### Detailed Description

The Get-SCOMMaintenanceMode cmdlet gets details about maintenance mode entries.

When you use the Get-SCOMMaintenance mode cmdlet to retrieve information about maintenance mode entries, the start time, end time, and scheduled end time values are returned in the computer's local time. If you pipe the results of Get-SCOMMaintenance mode to Format-Table, the start time, end time, and scheduled end time values are returned in UTC time.

### Parameters

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

Specifies the name of a computer to establish a connection with. The computer must be running the System Center Data Access Service. The default value is the computer for the current management group connection. Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

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



## **-Credential<PSCredential>**

Specifies a user account under which the management group connection will run. The default is the current user. Type a user name, such as "User01", "Domain01\User01", or "User@Domain.com", or enter a PSCredential object, such as one returned by the Get-Credential cmdlet. When you type a user name, you will be prompted for a password. For more information, type Get-Help Get-Credential.

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

## **-History**

Indicates that all maintenance mode entries will be returned whether or not they are active. By default, only active maintenance mode entries are returned.

Aliases	none
Required?	false
Position?	2
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-Instance<MonitoringObject[]>**

Returns maintenance mode entries for the specified class instance object. Enter a variable that represents the class instance, or type a command that gets the class instance. This parameter also accepts group objects.

For information about how to get a class instance object, type Get-Help Get-SCOMClassInstance.

Aliases	none
Required?	false
Position?	1
Default Value	none

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

## **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection. Specify a Management group connection object, such as one returned by the Get-SCOMManagementGroupConnection cmdlet.

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

## **<CommonParameters>**

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

## **Examples**

### **----- EXAMPLE 1 -----**

Description

-----

This command gets all active maintenance mode entries.

```
PS C:\>Get-SCOMMaintenanceMode
```

### **----- EXAMPLE 2 -----**

Description

-----

This command gets all active maintenance mode entries for class instances in the Contoso.com domain. If the command encounters a class instance for which there is no active maintenance mode entry, the SilentlyContinue value for the ErrorAction parameter allows the command to continue without displaying an error.

```
PS C:\>Get-SCOMMaintenanceMode -Instance (Get-SCOMClassInstance -Name *.Contoso.com) -
ErrorAction SilentlyContinue
```

### ----- EXAMPLE 3 -----

Description

-----

This command gets all class instances in the Contoso.com domain and then uses the pipeline operator (|) to pass the class instance objects to the Get-SCOMMaintenanceMode cmdlet. Because the History parameter is used, all maintenance mode entries are returned for the class instance objects, regardless of whether they are active.

```
PS C:\>Get-SCOMClassInstance -Name *.Contoso.com | Get-SCOMMaintenanceMode -History
```

## Related topics

[Get-SCOMClassInstance](#)

[Set-SCOMMaintenanceMode](#)

[Start-SCOMMaintenanceMode](#)

# Get-SCOMManagementGroup

---

## Get-SCOMManagementGroup

Gets a Microsoft.EnterpriseManagement.ManagementGroup object for the current management group.

### Syntax

Parameter Set: Empty

```
Get-SCOMManagementGroup [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCOMManagementGroup cmdlet gets a Microsoft.EnterpriseManagement.ManagementGroup object for the current management group. The ManagementGroup object is used as the base for all Operations Manager SDK operations.

### Parameters

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

Specifies a computer to establish a connection with. The computer must be running the System Center Data Access service. The default is the computer for the current management group connection.

Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

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

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

Specifies a user account under which the management group connection will run. It must have access to the server that is specified in the ComputerName parameter if specified. The default is the current user.

Enter a PSCredential object, such as one that is returned by the Get-Credential cmdlet.

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

## **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection.

Enter a Management group connection object, such as one returned by the Get-SCManagementGroupConnection cmdlet.

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

## **<CommonParameters>**

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

## **Examples**

### **----- EXAMPLE 1 -----**

This command gets a Microsoft.EnterpriseManagement.ManagementGroup object for the current management group.

```
PS C:\>Get-SCOMManagementGroup
```

## ----- EXAMPLE 2 -----

This example gets a management group object and then pipes it to the Get-Member cmdlet in order to display all public properties and methods available for use.

```
PS C:\>$mg = Get-SCOMManagementGroup
```

```
PS C:\>$mg | Get-Member
```

# Get-SCOMManagementGroupConnection

---

## Get-SCOMManagementGroupConnection

Retrieves all management group connections, including the IsActive state of these connections. Only one connection will have its IsActive state set to True because only one connection can be active at any time.

### Syntax

Parameter Set: FromComputerNames

```
Get-SCOMManagementGroupConnection [-ComputerName] <String[]> [ <CommonParameters>]
```

Parameter Set: FromInstanceId

```
Get-SCOMManagementGroupConnection [-Id] <Guid[]> [ <CommonParameters>]
```

Parameter Set: FromManagementGroupName

```
Get-SCOMManagementGroupConnection [-ManagementGroupName] <String[]> [ <CommonParameters>]
```

### Detailed Description

The Get-SCOMManagementGroupConnection cmdlet retrieves all management group connections, including the IsActive state of these connections. Only one connection will have its IsActive state set to True, because only one connection can be active at any time.

### Parameters

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

Specifies a computer to establish a connection with. The computer must be running the System Center Data Access service. The default value is the computer for the current management group connection. Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name (FQDN). To specify the local computer, type the computer name, "localhost", or a dot (.).

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

## **-Id<Guid[]>**

Specifies the ID of a management group for which to retrieve connections.

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

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

Specifies the names of the management groups for which to retrieve connections.

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

## **<CommonParameters>**

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

## **Inputs**

The input type is the type of the objects that you can pipe to the cmdlet.

- **System.String** You can pipe a computer name to the ComputerName parameter of the Get-SCOMManagementGroupConnection cmdlet.

## **Outputs**

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

- **SessionObject** This cmdlet generates a SessionObject object.



## Examples

### ----- EXAMPLE 1 -----

The command in this example retrieves all the available management group connections.

```
PS C:\>Get-SCOMManagementGroupConnection
```

## Related topics

[New-SCOMManagementGroupConnection](#)

[Remove-SCOMManagementGroupConnection](#)

[Set-SCOMManagementGroupConnection](#)

# Get-SCOMManagementPack

---

## Get-SCOMManagementPack

Retrieves management packs that have been imported into a management group; that exist in management pack files (.mp, .xml); or that reside within a management pack bundle (.mpb) file.

### Syntax

Parameter Set: \_\_AllParameterSets

```
Get-SCOMManagementPack [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromManagementPackBundle

```
Get-SCOMManagementPack [-BundleFile] <String[]> [ <CommonParameters>]
```

Parameter Set: FromManagementPackDisplayName

```
Get-SCOMManagementPack [-DisplayName] <String[]> [[-Recurse]] [ <CommonParameters>]
```

Parameter Set: FromManagementPackFile

```
Get-SCOMManagementPack [-ManagementPackFile] <String[]> [ <CommonParameters>]
```

Parameter Set: FromManagementPackGuid

```
Get-SCOMManagementPack [-Id] <Guid[]> [[-Recurse]] [ <CommonParameters>]
```

Parameter Set: FromManagementPackName

```
Get-SCOMManagementPack [-Name] <String[]> [[-Recurse]] [ <CommonParameters>]
```

### Detailed Description

The Get-SCOMManagementPack cmdlet retrieves management packs that have been imported into a management group; that exist in management pack files (.mp, .xml); or that reside within a management pack bundle (.mpb) file.

### Parameters

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

Specifies the names of the management pack bundle files (.mpb) from which to retrieve management packs.

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

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

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

Specifies a computer to establish a connection with. The computer must be running the System Center Data Access service. The default value is the computer for the current management group connection. Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name (FQDN). To specify the local computer, type the computer name, "localhost", or a dot (.).

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

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

Specifies a user account under which the management group connection will run. The account must have access to the server that is specified in the ComputerName parameter, if the server is specified. The default value is the current user.

You can enter a PSCredential object that is returned by the Get-Credential cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	The user account of the current context
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

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

Specifies the display name of the management packs to retrieve. This parameter is interpreted as a regular expression.

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

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

**-Id<Guid[]>**

Specifies the Id of the management packs to retrieve.

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

**-ManagementPackFile<String[]>**

Specifies the file name (.xml or .mp) of the management packs to retrieve.

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

**-Name<String[]>**

Specifies the names of the management packs to retrieve. This parameter is interpreted as a regular expression.

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

Required?	true
Position?	1
Default Value	.*
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-Recurse**

Recursively retrieves all management packs that depend on the management packs that are specified to be retrieved.

Aliases	none
Required?	false
Position?	2
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-SCSession<Connection[]>**

Specifies a connection to a management server. The default value is the current management group connection.

You can enter a management group connection object that is returned by the Get-SCOMManagementGroupConnection cmdlet.

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

## **<CommonParameters>**

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

## Inputs

The input type is the type of the objects that you can pipe to the cmdlet.

- **System.Guid**You can pipe the Guid of a management pack to the Id parameter of the Get-SCOMManagementPack cmdlet.
- **System.String**You can pipe the name of a management pack to the Name parameter of the Get-SCOMManagementPack cmdlet.

## Outputs

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

- **Microsoft.EnterpriseManagement.Configuration.ManagementPack**The management pack object contains the management pack and its properties.

## Examples

### ----- EXAMPLE 1 -----

The command in this example retrieves all imported management packs.

```
PS C:\>Get-SCOMManagementPack
```

### ----- EXAMPLE 2 -----

The command in this example retrieves only the System.Library management pack.

```
PS C:\>Get-SCOMManagementPack -Name System.Library
```

## Related topics

[Import-SCOMManagementPack](#)

[New-SCOMManagementPackBundle](#)

[Export-SCOMManagementPack](#)

[New-SCOMManagementPack](#)

[Protect-SCOMManagementPack](#)

[Remove-SCOMManagementPack](#)

[Test-SCOMManagementPack](#)

# Get-SCOMManagementServer

---

## Get-SCOMManagementServer

Gets the management servers in the management group.

### Syntax

Parameter Set: Empty

```
Get-SCOMManagementServer [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromManagementServerNames

```
Get-SCOMManagementServer [[-Name] <String[]> ] [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCOMManagementServer cmdlet gets the management servers in the management group.

### Parameters

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

Specifies the name of a computer to establish a connection with. The computer must be running the System Center Data Access Service. The default value is the computer for the current management group connection. Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

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

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

Specifies a user account under which the management group connection will run. The default is the current user. Type a user name, such as "User01", "Domain01\User01", or "User@Domain.com", or

enter a PSCredential object, such as one returned by the Get-Credential cmdlet. When you type a user name, you will be prompted for a password. For more information, type Get-Help Get-Credential.

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

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

Specifies the name of an object.

This parameter takes one or more strings and will look for matches in the Name property of the given objects that this cmdlet works with.

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

### **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection. Specify a Management group connection object, such as one returned by the Get-SCOMManagementGroupConnection cmdlet.

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



## <CommonParameters>

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

## Examples

### ----- EXAMPLE 1 -----

Description

-----

This command gets the management servers in the active connection that are named Server01.Contoso.com and that have a domain of Contoso.com.

```
PS C:\>Get-SCOMManagementServer -Name "Server01.Contoso.com", "*.Contoso.com"
```

### ----- EXAMPLE 2 -----

Description

-----

This command establishes a temporary connection to Server01.Contoso.com and gets all management servers in that management group.

```
PS C:\>Get-SCOMManagementServer -Computername "Server01.Contoso.com"
```

## Related topics

[Get-SCOMGatewayManagementServer](#)

# Get-SCOMMonitor

---

## Get-SCOMMonitor

Gets the specified monitors.

### Syntax

Parameter Set: Empty

```
Get-SCOMMonitor [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromEMO

```
Get-SCOMMonitor [-Instance] <EnterpriseManagementObject[]> [[-Recurse]] [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromManagementPack

```
Get-SCOMMonitor [-ManagementPack] <ManagementPack[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromManagementPackClass

```
Get-SCOMMonitor [-Target] <ManagementPackClass[]> [[-Recurse]] [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromMonitoringDisplayRuleName

```
Get-SCOMMonitor [-DisplayName] <String[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromMonitoringRuleId

```
Get-SCOMMonitor -Id <Guid[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromMonitorName

```
Get-SCOMMonitor -Name <String[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

## Detailed Description

The Get-SCOMMonitor cmdlet gets the specified monitors.

## Parameters

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

Specifies the name of a computer to establish a connection with. The computer must be running the System Center Data Access Service. The default value is the computer for the current management group connection. Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

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

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

Specifies a user account under which the management group connection will run. The default is the current user. Type a user name, such as "User01", "Domain01\User01", or "User@Domain.com", or enter a PSCredential object, such as one returned by the Get-Credential cmdlet. When you type a user name, you will be prompted for a password. For more information, type Get-Help Get-Credential.

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

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

Specifies the display name of an object. This parameter takes one or more strings and will look for matches in the DisplayName property of the given objects that this cmdlet works with. DisplayName values may vary depending on which localized management packs are imported into the management group and the locale of the user running Windows PowerShell.

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

## **-Id<Guid[]>**

Retrieves the monitor with the specified GUID.

The GUID is stored in the Id property of the object that represents an SCOMMonitor. To get the GUID of a monitor, type "Get-SCOMMonitor | Format-Table DisplayName, Id".

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

## **-Instance<EnterpriseManagementObject[]>**

Retrieves monitors for the specified class instance object. Enter a variable that represents the class instance, or type a command that gets the class instance. This parameter also accepts group objects.

For information about how to get a class instance object, type Get-Help Get-SCOMClassInstance.

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

## **-ManagementPack<ManagementPack[]>**

Retrieves monitors for one or more specified management pack objects. Enter a variable that represents the management packs, or type a command that gets the management packs.

For information about how to get a management pack object, type Get-Help Get-SCOMManagementPack.

Aliases	none
Required?	true
Position?	1
Default Value	none

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

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

Specifies the name of an object.

This parameter takes one or more strings and will look for matches in the Name property of the given objects that this cmdlet works with.

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

### **-Recurse**

Aliases	none
Required?	false
Position?	2
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection. Specify a Management group connection object, such as one returned by the Get-SCOMManagementGroupConnection cmdlet.

Aliases	none
Required?	false
Position?	named

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

## **-Target<ManagementPackClass[]>**

Retrieves monitors for one or more target class objects. Enter a variable that represents the target classes, or type a command that gets the target classes.

For information about how to get a class object, type Get-Help Get-SCOMClass.

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

## **<CommonParameters>**

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

## **Examples**

### **----- EXAMPLE 1 -----**

Description

-----

This command gets all monitors that have a name beginning with "System.Health".

```
PS C:\>Get-SCOMMonitor -Name System.Health*
```

### **----- EXAMPLE 2 -----**

Description

-----

This command gets all monitors that have a display name containing "Performance".

```
PS C:\>Get-SCOMMonitor -DisplayName *Performance*
```

### ----- EXAMPLE 3 -----

#### Description

-----

This command gets the management pack with the display name of "System Center Core Library" and then uses the pipeline operator (|) to pass the management pack object to the Get-SCOMMonitor cmdlet which gets all monitors for the management pack object.

```
PS C:\>Get-SCOMManagementPack -DisplayName "System Center Core Library" | Get-SCOMMonitor
```

### ----- EXAMPLE 4 -----

#### Description

-----

The first command gets the path to a management pack file and saves it in the \$MPFile variable.

The second command uses the ManagementPack parameter to get the monitors contained in the management pack stored in the \$MPFile variable.

```
PS C:\>$MPFile = "C:\Program Files\System Center Operations Manager  
2007\Microsoft.SystemCenter.2007.mp"
```

```
PS C:\>Get-SCOMMonitor -ManagementPack $MpFile
```

### ----- EXAMPLE 5 -----

#### Description

-----

The first command gets all classes with "health" in their display name and uses the pipeline operator (|) to pass the class objects to the Get-SCOMClassInstance cmdlet which gets the instance for each class, and then stores the results in the \$Instances variable.

The second command gets the monitors for the class instances stored in the \$Instances variable.

```
PS C:\>$Instances = Get-SCOMClass -DisplayName *health* | Get-SCOMClassInstance
```

```
PS C:\>Get-SCOMMonitor -Instance $Instances
```

### ----- EXAMPLE 6 -----

#### Description

-----

This command gets the monitors with target classes that contain "health" in their display name.

```
PS C:\>Get-SCOMMonitor -Target (Get-SCOMClass -DisplayName *health*)
```

### ----- EXAMPLE 7 -----

#### Description

-----

This command gets the monitor that has an Id of 7413b06b-a95b-4ae3-98f2-dac9ff76dabd.

```
PS C:\>Get-SCOMMonitor -Id 7413b06b-a95b-4ae3-98f2-dac9ff76dabd
```

## Related topics

[Get-SCOMClassInstance](#)

[Get-SCOMManagementPack](#)



# Get-SCOMNotificationChannel

---

## Get-SCOMNotificationChannel

Gets the notification channels defined in the management group.

### Syntax

Parameter Set: Empty

```
Get-SCOMNotificationChannel [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromDisplayName

```
Get-SCOMNotificationChannel [-DisplayName] <String[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCOMNotificationChannel cmdlet gets the notification channels defined in the management group.

### Parameters

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

Specifies a computer to establish a connection with. The computer must be running the System Center Data Access service. The default is the computer for the current management group connection.

Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

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

## **-Credential<PSCredential>**

Specifies a user account under which the management group connection will run. It must have access to the server that is specified in the ComputerName parameter if specified. The default is the current user.

Enter a PSCredential object, such as one that is returned by the Get-Credential cmdlet.

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

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

The display name of the channel.

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

## **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection.

Enter a Management group connection object, such as one returned by the Get-SCManagementGroupConnection cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none

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

## <CommonParameters>

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

## Examples

### ----- EXAMPLE 1 -----

This command gets all notification channels.

```
PS C:\>Get-SCOMNotificationChannel
```

### ----- EXAMPLE 2 -----

This command gets the notification channel named "Email channel".

```
PS C:\>Get-SCOMNotificationChannel -DisplayName "Email channel"
```

### ----- EXAMPLE 3 -----

This command gets the notification channel named "MyCustomChannel".

```
PS C:\>Get-SCOMNotificationChannel "MyCustomChannel"
```

# Get-SCOMNotificationSubscriber

---

## Get-SCOMNotificationSubscriber

Gets the notification subscribers defined in the management group.

### Syntax

Parameter Set: Empty

```
Get-SCOMNotificationSubscriber [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromId

```
Get-SCOMNotificationSubscriber -Id <Guid[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromName

```
Get-SCOMNotificationSubscriber [-Name] <String[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCOMNotificationSubscriber cmdlet gets the notification subscribers defined in the management group.

### Parameters

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

Specifies a computer to establish a connection with. The computer must be running the System Center Data Access service. The default is the computer for the current management group connection.

Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

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

## **-Credential<PSCredential>**

Specifies a user account under which the management group connection will run. It must have access to the server that is specified in the ComputerName parameter if specified. The default is the current user.

Enter a PSCredential object, such as one that is returned by the Get-Credential cmdlet.

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

## **-Id<Guid[]>**

The ID of a subscriber.

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

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

The name of a subscriber.

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

## **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection.

Enter a Management group connection object, such as one returned by the Get-SCManagementGroupConnection cmdlet.

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

## **<CommonParameters>**

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

## **Examples**

### **----- EXAMPLE 1 -----**

This command gets all notification subscribers.

```
PS C:\>Get-SCOMNotificationSubscriber
```

### **----- EXAMPLE 2 -----**

This command gets the notification subscriber with the name "Katarina".

```
PS C:\>Get-SCOMNotificationSubscriber -Name "Katarina"
```

# Get-SCOMNotificationSubscription

---

## Get-SCOMNotificationSubscription

Gets a list of notification subscriptions.

### Syntax

Parameter Set: Empty

```
Get-SCOMNotificationSubscription [-ComputerName <String[]> ] [-Credential <PSCredential> ]  
[-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromDisplayName

```
Get-SCOMNotificationSubscription [-DisplayName] <String[]> [-ComputerName <String[]> ] [-  
Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromId

```
Get-SCOMNotificationSubscription -Id <Guid[]> [-ComputerName <String[]> ] [-Credential  
<PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromName

```
Get-SCOMNotificationSubscription -Name <String[]> [-ComputerName <String[]> ] [-Credential  
<PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCOMNotificationSubscription cmdlet gets a list of notification subscriptions.

### Parameters

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

Specifies the name of a computer to establish a connection with. The computer must be running the System Center Data Access Service. The default value is the computer for the current management group connection. Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

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

## **-Credential<PSCredential>**

Specifies a user account under which the management group connection will run. The default is the current user. Type a user name, such as "User01", "Domain01\User01", or "User@Domain.com", or enter a PSCredential object, such as one returned by the Get-Credential cmdlet. When you type a user name, you will be prompted for a password. For more information, type Get-Help Get-Credential.

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

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

Specifies the display name of an object. This parameter takes one or more strings and will look for matches in the DisplayName property of the given objects that this cmdlet works with. DisplayName values may vary depending on which localized management packs are imported into the management group and the locale of the user running Windows PowerShell.

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

## **-Id<Guid[]>**

Specifies a Guid. This parameter takes one or more globally unique identifiers (Guids) and uses them to compare against the id property of the objects retrieved by this cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none



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

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

Specifies a name. This parameter takes one or more strings and will look for matches in the Name property of the given objects that this cmdlet works with.

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

### **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection. Specify a Management group connection object, such as one returned by the Get-SCOMManagementGroupConnection cmdlet.

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

### **<CommonParameters>**

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

## Examples

### ----- EXAMPLE 1 -----

This command gets the notification subscription with the displayname of subscription01 and displays information about the subscription to the user.

```
PS C:\> Get-SCOMNotificationSubscription -DisplayName "subscription01"
```

# Get-SCOMOverride

---

## Get-SCOMOverride

Gets a list of overrides or a resulting set of overrides.

### Syntax

Parameter Set: Empty

```
Get-SCOMOverride [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromManagementPackDiagnostic

```
Get-SCOMOverride [[-Diagnostic] <ManagementPackDiagnostic[]> ] [-Class <ManagementPackClass[]> ] [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-Group <EnterpriseManagementObject[]> ] [-Instance <EnterpriseManagementObject[]> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromManagementPackDiscovery

```
Get-SCOMOverride [[-Discovery] <ManagementPackDiscovery[]> ] [-Class <ManagementPackClass[]> ] [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-Group <EnterpriseManagementObject[]> ] [-Instance <EnterpriseManagementObject[]> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromManagementPackMonitor

```
Get-SCOMOverride [[-Monitor] <ManagementPackMonitor[]> ] [-Class <ManagementPackClass[]> ] [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-Group <EnterpriseManagementObject[]> ] [-Instance <EnterpriseManagementObject[]> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromManagementPackRecovery

```
Get-SCOMOverride [[-Recovery] <ManagementPackRecovery[]> ] [-Class <ManagementPackClass[]> ] [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-Group <EnterpriseManagementObject[]> ] [-Instance <EnterpriseManagementObject[]> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromManagementPackRule

```
Get-SCOMOverride [[-Rule] <ManagementPackRule[]> ] [-Class <ManagementPackClass[]> ] [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-Group <EnterpriseManagementObject[]> ] [-Instance <EnterpriseManagementObject[]> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromTask

```
Get-SCOMOverride [[-Task] <ManagementPackTask[]> ] [-Class <ManagementPackClass[]> ] [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-Group <EnterpriseManagementObject[]> ] [-Instance <EnterpriseManagementObject[]> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCOMOverride cmdlet gets a list of overrides or a resulting set of overrides.

## Parameters

### **-Class<ManagementPackClass[]>**

Retrieves overrides for one or more specified class objects. Enter a variable that represents the classes, or type a command that gets the classes. For information about how to get a class object, type Get-Help Get-SCOMClass.

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

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

Specifies the name of a computer to establish a connection with. The computer must be running the System Center Data Access Service. The default value is the computer for the current management group connection. Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

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

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

Specifies a user account under which the management group connection will run. The default is the current user. Type a user name, such as "User01", "Domain01\User01", or "User@Domain.com", or enter a PSCredential object, such as one returned by the Get-Credential cmdlet. When you type a user name, you will be prompted for a password. For more information, type Get-Help Get-Credential.

Aliases	none
Required?	false

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

### **-Diagnostic<ManagementPackDiagnostic[]>**

Retrieves overrides for one or more specified diagnostic objects. Enter a variable that represents the diagnostics, or type a command that gets the diagnostics.

For information about how to get a diagnostic object, type Get-Help Get-SCOMDiagnostic.

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

### **-Discovery<ManagementPackDiscovery[]>**

Retrieves overrides for one or more specified discovery objects. Enter a variable that represents the discoveries, or type a command that gets the discoveries.

For information about how to get a discovery object, type Get-Help Get-SCOMDiscovery.

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

### **-Group<EnterpriseManagementObject[]>**

Retrieves overrides for one or more group objects. Enter a variable that represents the groups, or type a command that gets the groups.

For information about how to get a group object, type Get-Help Get-SCOMGroup.

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

**-Instance<EnterpriseManagementObject[]>**

Retrieves overrides for one or more specified class instance objects. Enter a variable that represents the class instances, or type a command that gets the class instances. This parameter also accepts group objects.

For information about how to get a class instance object, type Get-Help Get-SCOMClassInstance.

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

**-Monitor<ManagementPackMonitor[]>**

Retrieves overrides for one or more specified monitor objects. Enter a variable that represents the monitors, or type a command that gets the monitors.

For information about how to get a monitor object, type Get-Help Get-SCOMMonitor.

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

## **-Recovery<ManagementPackRecovery[]>**

Retrieves overrides for one or more recovery objects. Enter a variable that represents the recoveries, or type a command that gets the recoveries.

For information about how to get a recovery object, type Get-Help Get-SCOMRecovery.

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

## **-Rule<ManagementPackRule[]>**

Retrieves overrides for one or more specified monitoring rule objects. Enter a variable that represents the monitoring rules, or type a command that gets the monitoring rules.

For information about how to get a monitoring rule object, type Get-Help Get-SCOMRule.

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

## **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection. Specify a Management group connection object, such as one returned by the Get-SCOMManagementGroupConnection cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none

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

## **-Task<ManagementPackTask[]>**

Retrieves overrides for one or more specified task objects. Enter a variable that represents the tasks, or type a command that gets the tasks.

For information about how to get a task object, type Get-Help Get-SCOMTask.

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

## **<CommonParameters>**

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

## **Examples**

### **----- EXAMPLE 1 -----**

Description

-----

This command gets all monitoring rules that contain "health" in their name, and then returns the overrides for the rule objects.

Using the ErrorAction parameter with the SilentlyContinue value allows the command to continue if it cannot find an override for a specific monitoring rule, and will not display an error.

```
PS C:\>Get-SCOMRule -Name *health* | Get-SCOMOverride -ErrorAction SilentlyContinue
```

### **----- EXAMPLE 2 -----**

Description

-----

This command gets the override for monitors that contain "health" in their name and are in a class with a display name that contains "computer".



```
PS C:\>Get-SCOMOverride -Monitor (Get-SCOMMonitor -Name *health*) -Class (Get-SCOMClass -
DisplayName *computer*)
```

### ----- EXAMPLE 3 -----

Description

-----

This command gets all discoveries that contain "system" in their name, and then returns the overrides for those discovery objects.

Using the ErrorAction parameter with the SilentlyContinue value allows the command to continue if it cannot find an override for a specific discovery, and will not display an error.

```
PS C:\>Get-SCOMDiscovery -name *system* | Get-SCOMOverride -ErrorAction SilentlyContinue
```

## Related topics

[Get-SCOMClass](#)

[Get-SCOMClassInstance](#)

[Get-SCOMDiscovery](#)

[Get-SCOMGroup](#)

[Get-SCOMDiagnostic](#)

[Get-SCOMMonitor](#)

[Get-SCOMRecovery](#)

[Get-SCOMRule](#)

[Get-SCOMTask](#)

# Get-SCOMOverrideResult

---

## Get-SCOMOverrideResult

Gets override results.

### Syntax

Parameter Set: FromClassDiscovery

```
Get-SCOMOverrideResult [-Class] <ManagementPackClass[]> [-Discovery]  
<ManagementPackDiscovery[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-  
SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromClassMonitor

```
Get-SCOMOverrideResult [-Class] <ManagementPackClass[]> [-Monitor] <ManagementPackMonitor[]>  
[-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [  
<CommonParameters>]
```

Parameter Set: FromClassRule

```
Get-SCOMOverrideResult [-Class] <ManagementPackClass[]> [-Rule] <ManagementPackRule[]> [-  
ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [  
<CommonParameters>]
```

Parameter Set: FromInstanceDiscovery

```
Get-SCOMOverrideResult [-Instance] <EnterpriseManagementObject[]> [-Discovery]  
<ManagementPackDiscovery[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-  
SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromInstanceMonitor

```
Get-SCOMOverrideResult [-Instance] <EnterpriseManagementObject[]> [-Monitor]  
<ManagementPackMonitor[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-  
SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromInstanceRule

```
Get-SCOMOverrideResult [-Instance] <EnterpriseManagementObject[]> [-Rule]  
<ManagementPackRule[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession  
<Connection[]> ] [ <CommonParameters>]
```

## Detailed Description

The Get-SCOMOverrideResult cmdlet gets existing overrides associated with specified workflows, scoped to a specified class or class instance.

## Parameters

### **-Class<ManagementPackClass[]>**

Specifies a management pack class object. Overrides scoped to this class are retrieved.

For information about how to get a management pack class object, type: Get-Help Get-SCOMClass.

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

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

Specifies a computer to establish a connection with. The computer must be running the Operations Manager Data Access service.

Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

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

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

Specifies a user account under which the management group connection will run. The default is the current user. Type a user name, such as "User01", "Domain01\User01", or "User@Domain.com", or enter a PSCredential object, such as one returned by the Get-Credential cmdlet. When you type a user name, you will be prompted for a password. For more information, type Get-Help Get-Credential.

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

## **-Discovery<ManagementPackDiscovery[]>**

Specifies the discovery workflow object for which overrides are retrieved.

For information about how to get a discovery object, type: Get-Help Get-SCOMDiscovery.

Aliases	none
Required?	true
Position?	2
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-Instance<EnterpriseManagementObject[]>**

Specifies an instance of a class. Overrides scoped to this instance are retrieved.

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

## **-Monitor<ManagementPackMonitor[]>**

Specifies the monitor workflow object for which overrides are retrieved.

For information about how to get a monitor object, type: Get-Help Get-SCOMMonitor.

Aliases	none
Required?	true
Position?	2
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-Rule<ManagementPackRule[]>**

Specifies the monitoring rule workflow object for which overrides are retrieved.

For information about how to get a rule object, type: Get-Help Get-SCOMRule.

Aliases	none
Required?	true
Position?	2
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection.

For information about how to get a management group connection, type: Get-Help Get-SCOMManagementGroupConnection cmdlet.

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

## **<CommonParameters>**

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

## **Examples**

### **----- EXAMPLE 1 -----**

Description

-----

The first command gets the class object named Memory and stores the object in the \$Class variable.

The second command gets the discovery object with the display name "Discover Windows Server Computers" and stores the object in the \$Discovery variable.

The third command gets the management group connection object for Server01 and stores the object in the \$Session variable.

The last command gets the list of overrides for the specified discovery, scoped to the specified class, for the specified management group connection, and returns information about the override result to the user.

```
PS C:\> $Class = Get-SCOMClass -DisplayName "Memory"
PS C:\>$Discovery = Get-SCOMDiscovery -DisplayName "Discover Windows Server Computers"
PS C:\>$Session = Get-SCOMManagementGroupConnection -ComputerName "Server01.Contoso.com"
PS C:\>Get-SCOMOverrideResult -Class $Class -Discovery $Discovery -Session $Session
```

## ----- EXAMPLE 2 -----

Description

-----

The first command gets the class instance object named Operations Manager Agents and stores the object in the \$Instance variable.

The second command gets the monitor object with the display name "Management Service connectivity state" and stores the object in the \$Monitor variable.

The last command gets the override result for the specified instance and monitor, and returns information about the override result to the user.

```
PS C:\> $Instance = Get-SCOMClassInstance -DisplayName " Operations Manager Agents"
PS C:\>$Monitor = Get-SCOMMonitor -DisplayName "Management Service connectivity state"
PS C:\>Get-SCOMOverrideResult -Instance $Instance -Monitor $Monitor
```

## ----- EXAMPLE 3 -----

Description

-----

The first command gets the monitoring rule object with the display name "Alert on Failed Power Shell Scripts" and stores the object in the \$Rule variable.

The second command gets the class object named Memory and uses the pipeline operator to pass the object to Get-SCOMOverrideResult which gets the override result for the specified rule, scoped to the piped class object.

```
PS C:\> $Rule = Get-SCOMRule -DisplayName "Alert on Failed Power Shell Scripts"
PS C:\>Get-SCOMClass -DisplayName "Memory" | Get-SCOMOverrideResult -Rule $Rule
```

## ----- EXAMPLE 4 -----

Description

-----

The first command gets the discovery object with the display name "Discover Windows Server Computers" and stores the object in the \$Discovery variable.

The second command gets the class instance object named Operations Manager Agents and uses the pipeline operator to pass the object to Get-SCOMOverrideResult which gets the override result for the specified discovery, scoped to the piped class instance object.

```
PS C:\> $Discovery = Get-SCOMDiscovery -DisplayName "Discover Windows Server Computers"
```

```
PS C:\>Get-SCOMClassInstance -DisplayName " Operations Manager Agents" | Get-SCOMOverrideResult -Discovery $Discovery
```

# Get-SCOMPparentManagementServer

---

## Get-SCOMPparentManagementServer

Gets the management servers that an agent reports to.

### Syntax

Parameter Set: FromAgent

```
Get-SCOMPparentManagementServer [-Agent] <AgentManagedComputer[]> [ <CommonParameters>]
```

Parameter Set: FromGatewayManagementServer

```
Get-SCOMPparentManagementServer [-GatewayServer] <ManagementServer[]> [ <CommonParameters>]
```

### Detailed Description

The Get-SCOMPparentManagementServer cmdlet gets the management servers that an agent reports to.

### Parameters

#### **-Agent<AgentManagedComputer[]>**

Specifies one or more agent objects. Enter a variable that represents the agents, or type a command that gets the agents.

For information about how to get an agent object, type Get-Help Get-SCAgent.

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

#### **-GatewayServer<ManagementServer[]>**

Specifies a gateway management server.

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



Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	true (ByValue, 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 -----

This command gets the management servers that the agent named Server01 reports to.

The command in parenthesis, which is executed first, gets the agent named Server01. The cmdlet then uses the InputObject parameter to pass the results of the command in parentheses to Get-SCOMPparentManagementServer. Get-SCOMPparentManagementServer then retrieves the management servers that the agent reports to.

```
PS C:\>Get-SCOMPparentManagementServer -Agent (Get-SCAgent -Name "Server01.Contoso.com")
```

### ----- EXAMPLE 2 -----

The first command gets agents that begin with "Server" in the Contoso.com domain by piping the name with a wildcard character into the Get-SCAgent cmdlet. The command stores the agents in the \$Agents variable.

The second command gets the management servers that the agents stored in the \$Agents variable report to.

```
PS C:\>$Agents = "Server*.Contoso.com" | Get-SCAgent
```

```
PS C:\>Get-SCOMPparentManagementServer -Agent $Agents
```

### ----- EXAMPLE 3 -----

This command gets the agent named "Server01.Consoso.com" and returns the management servers that the agent reports to.

```
PS C:\>"Server01.Contoso.com" | Get-SCAgent | Get-SCOMPparentManagementServer
```

# Get-SCOMPendingManagement

---

## Get-SCOMPendingManagement

Gets pending agent management actions.

### Syntax

Parameter Set: Empty

```
Get-SCOMPendingManagement [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCOMPendingManagement gets pending agent management actions.

### Parameters

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

Specifies the name of a computer to establish a connection with. The computer must be running the System Center Data Access Service. The default value is the computer for the current management group connection. Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

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

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

Specifies a user account under which the management group connection will run. The default is the current user. Type a user name, such as "User01", "Domain01\User01", or "User@Domain.com", or enter a PSCredential object, such as one returned by the Get-Credential cmdlet. When you type a user name, you will be prompted for a password. For more information, type Get-Help Get-Credential.

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

## **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection. Specify a Management group connection object, such as one returned by the Get-SCOMManagementGroupConnection cmdlet.

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

## **<CommonParameters>**

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

## **Examples**

### **----- EXAMPLE 1 -----**

Description

-----

This command gets all pending management entries and sorts them by AgentName.

```
PS C:\>Get-SCOMPendingManagement | Sort AgentName
```

### **----- EXAMPLE 2 -----**

Description

-----

This command gets the agent management entries that are pending with an action of "ManualApproval", and sorts the returned entries by AgentName.

```
PS C:\>Get-SCOMPendingManagement | where {$_.AgentPendingActionType -eq "ManualApproval"} | Sort AgentName
```

### ----- EXAMPLE 3 -----

Description

-----

This command gets all pending management entries, groups them by their pending action type, and then sorts them in descending order by the number of entries there are per action type group. It then returns the action type group name and number of entries in each group.

```
PS C:\>Get-SCOMPendingManagement | Group AgentPendingActionType | Sort -Descending Count | Select-object Name, Count
```

## Related topics

[Approve-SCOMPendingManagement](#)

[Deny-SCOMPendingManagement](#)

# Get-SCOMRecovery

---

## Get-SCOMRecovery

Gets a list of recoveries.

### Syntax

Parameter Set: Empty

```
Get-SCOMRecovery [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromManagementPack

```
Get-SCOMRecovery -ManagementPack <ManagementPack[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromManagementPackClass

```
Get-SCOMRecovery -Target <ManagementPackClass[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromMonitorInstance

```
Get-SCOMRecovery -Monitor <ManagementPackMonitor[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromRecoveryDisplayName

```
Get-SCOMRecovery [-DisplayName] <String[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromRecoveryId

```
Get-SCOMRecovery -Id <Guid[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromRecoveryName

```
Get-SCOMRecovery -Name <String[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

## Detailed Description

The Get-SCOMRecovery cmdlet gets a list of recoveries.

## Parameters

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

Specifies the name of a computer to establish a connection with. The computer must be running the System Center Data Access Service. The default value is the computer for the current management group connection. Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

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

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

Specifies a user account under which the management group connection will run. The default is the current user. Type a user name, such as "User01", "Domain01\User01", or "User@Domain.com", or enter a PSCredential object, such as one returned by the Get-Credential cmdlet. When you type a user name, you will be prompted for a password. For more information, type Get-Help Get-Credential.

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

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

Specifies the display name of an object. This parameter takes one or more strings and will look for matches in the DisplayName property of the given objects that this cmdlet works with. DisplayName values may vary depending on which localized management packs are imported into the management group and the locale of the user running Windows PowerShell.

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

### **-Id<Guid[]>**

Specifies a Guid. This parameter takes one or more globally unique identifiers (Guids) and uses them to compare against the id property of the objects retrieved by this cmdlet.

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

### **-ManagementPack<ManagementPack[]>**

Specifies a management pack. This parameter takes one or more management pack objects. Retrieve management pack objects using the Get-SCManagementPack cmdlet.

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

### **-Monitor<ManagementPackMonitor[]>**

Specifies the monitors that this cmdlet should retrieve objects for.

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

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

Specifies a name. This parameter takes one or more strings and will look for matches in the Name property of the given objects that this cmdlet works with.

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

## **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection. Specify a Management group connection object, such as one returned by the Get-SCOMManagementGroupConnection cmdlet.

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

## **-Target<ManagementPackClass[]>**

Specifies the target classes to use to limit the results of this cmdlet. Retrieve classes using the Get-SCClass cmdlet.

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



## <CommonParameters>

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

## Examples

### ----- EXAMPLE 1 -----

This command gets all recoveries related to the Health Service.

```
PS C:\> Get-SCOMRecovery -Name Microsoft.SystemCenter.HealthService.Recovery.*
```

# Get-SCOMRelationship

---

## Get-SCOMRelationship

Retrieves information about relationship objects from Operations Manager.

### Syntax

Parameter Set: Empty

```
Get-SCOMRelationship [-Source] <ManagementPackClass[]> ] [[-Target] <ManagementPackClass[]> ] [ <CommonParameters>]
```

Parameter Set: \_\_AllParameterSets

```
Get-SCOMRelationship [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromManagementPack

```
Get-SCOMRelationship [-ManagementPack] <ManagementPack[]> [[-Source] <ManagementPackClass[]> ] [[-Target] <ManagementPackClass[]> ] [ <CommonParameters>]
```

Parameter Set: FromRelationshipDisplayName

```
Get-SCOMRelationship [-DisplayName] <String[]> [[-Source] <ManagementPackClass[]> ] [[-Target] <ManagementPackClass[]> ] [ <CommonParameters>]
```

Parameter Set: FromRelationshipId

```
Get-SCOMRelationship [-Id] <Guid[]> [ <CommonParameters>]
```

Parameter Set: FromRelationshipName

```
Get-SCOMRelationship [-Name] <String[]> [[-Source] <ManagementPackClass[]> ] [[-Target] <ManagementPackClass[]> ] [ <CommonParameters>]
```

## Detailed Description

The Get-SCOMRelationship cmdlet retrieves information about relationship objects from Operations Manager.

## Parameters

### -ComputerName<String[]>

Specifies a computer to establish a connection with. The computer must be running the System Center Data Access service. The default value is the computer for the current management group connection.

Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name (FQDN). To specify the local computer, type the computer name, "localhost", or a dot (.).

Aliases	none
Required?	false

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

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

Specifies a user account under which the management group connection will run. The account must have access to the server that is specified in the ComputerName parameter, if the server is specified. The default value is the current user.

You can enter a PSCredential object that is returned by the Get-Credential cmdlet.

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

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

Specifies the display name of the relationship object to retrieve.

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

### **-Id<Guid[]>**

Specifies the ID of the relationship object to retrieve. This may be a Guid or a string that will be converted to a Guid.

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

### **-ManagementPack<ManagementPack[]>**

Specifies one or more management packs containing the relationships to retrieve.

You can enter a ManagementPack object that is returned by the Get-SCOMManagementPack cmdlet.

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

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

Specifies the name of the relationship object to retrieve.

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

### **-SCSession<Connection[]>**

Specifies a connection to a management server. The default value is the current management group connection.

You can enter a management group connection object that is returned by the Get-SCOMManagementGroupConnection cmdlet.

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

### **-Source<ManagementPackClass[]>**

Specifies the classes that represent the source of the relationship. If more than one class is specified, any relationship that has one of those classes as a source is returned. The source class of the relationship must be an exact match of the specified class type.

If you specify classes for both the Target and the Source parameters, the cmdlet returns all relationships in which the target class is one of the specified target classes and the source class is one of the specified source classes.

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

### **-Target<ManagementPackClass[]>**

Specifies the classes that represent the target of the relationship. If more than one class is specified, any relationship that has one of those classes as a target is returned. The target class of the relationship must be an exact match of the specified class type.

If you specify classes for both the Target and the Source parameters, the cmdlet returns all relationships in which the target class is one of the specified target classes and the source class is one of the specified source classes.

Aliases	none
Required?	false

Position?	2
Default Value	none
Accept Pipeline Input?	true (ByValue)
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.

- **Microsoft.EnterpriseManagement.Configuration.ManagementPackClass** You can pipe a source class to the Source parameter of the Get-SCOMRelationship cmdlet.
- **Microsoft.EnterpriseManagement.Configuration.ManagementPackClass** You can pipe a target class to the Target parameter of the Get-SCOMRelationship cmdlet.
- **System.Guid** You can pipe a GUID to the Id parameter of the Get-SCOMRelationship cmdlet.
- **Microsoft.EnterpriseManagement.Configuration.ManagementPack** You can pipe a management pack to the ManagementPack parameter of the Get-SCOMRelationship cmdlet.
- **System.String** You can pipe a name to the Name parameter of the Get-SCOMRelationship cmdlet.

## Outputs

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

- This cmdlet does not generate any output.

## Examples

### ----- EXAMPLE 1 -----

The command in this example retrieves information about three relationship objects.

```
PS C:\>Get-SCOMRelationship | select-object -first 3
```

### ----- EXAMPLE 2 -----

The commands in this example retrieve relationships that target configuration items.

```
PS C:\>$x = Get-SCOMClass -name "system.configitem"
```

```
PS C:\>Get-SCOMRelationship -Target $x
```

### ----- EXAMPLE 3 -----

The commands in this example retrieve relationships that target configuration items, and any types that are derived from these configuration items.

```
PS C:\> $x = Get-SCOMClass -name "system.configitem"
```

```
PS C:\> $y = @($x; $x.GetDerivedTypes())
```

```
PS C:\> Get-SCOMRelationship -Target $y
```

# Get-SCOMRelationshipInstance

---

## Get-SCOMRelationshipInstance

Retrieves the instances of relationships from Operations Manager.

### Syntax

Parameter Set: \_\_AllParameterSets

```
Get-SCOMRelationshipInstance [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromRelationshipInstanceId

```
Get-SCOMRelationshipInstance [-Id] <Guid[]> [ <CommonParameters>]
```

Parameter Set: FromRelationshipInstanceSourceTarget

```
Get-SCOMRelationshipInstance [[-SourceInstance] <EnterpriseManagementObject[]> ] [[-TargetInstance] <EnterpriseManagementObject[]> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCOMRelationshipInstance cmdlet retrieves the instances of relationships from Operations Manager. These relationships describe the relationship of one ClassInstance to another ClassInstance.

### Parameters

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

Specifies a computer to establish a connection with. The computer must be running the System Center Data Access service. The default value is the computer for the current management group connection.

Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name (FQDN). To specify the local computer, type the computer name, "localhost", or a dot (.).

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



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

Specifies a user account under which the management group connection will run. The account must have access to the server that is specified in the ComputerName parameter, if the server is specified. The default value is the current user. You can enter a PSCredential object that is returned by the Get-Credential cmdlet.

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

### **-Id<Guid[]>**

Specifies the ID of the relationship object to retrieve. This may be a Guid or a string that will be converted to a Guid.

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

### **-SCSession<Connection[]>**

Specifies a connection to a management server. The default value is the current management group connection.

You can enter a management group connection object that is returned by the Get-SCOMManagementGroupConnection cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-SourceInstance<EnterpriseManagementObject[]>**

Specifies the instances that represent the source class of the relationships to be retrieved.

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

### **-TargetInstance<EnterpriseManagementObject[]>**

Specifies the instances that represent the target class of the relationships to be retrieved.

Aliases	none
Required?	false
Position?	2
Default Value	none
Accept Pipeline Input?	false
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.

- **System.Guid** You can pipe a GUID of a relationship object to the Id parameter of the Get-SCOMRelationshipInstance cmdlet.

## Outputs

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

- **EnterpriseManagementRelationshipObject**This cmdlet generates a relationship object.

## Examples

### ----- EXAMPLE 1 -----

The commands in this example retrieve all relationship instances for which the source instance and the target instance is the Health Service class.

```
PS C:\>$healthService = Get-SCOMClassInstance -Class (Get-SCOMClass -name  
Microsoft.SystemCenter.HealthService)
```

```
PS C:\>Get-SCOMRelationshipInstance -SourceInstance $healthService -TargetInstance  
$HealthService | format-Table
```

# Get-SCOMReportingSetting

---

## Get-SCOMReportingSetting

Gets the reporting server settings for the management group.

### Syntax

Parameter Set: Empty

```
Get-SCOMReportingSetting [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCOMReportingSetting cmdlet gets the reporting server settings for the management group.

### Parameters

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

Specifies a computer to establish a connection with. The computer must be running the System Center Data Access service. The default is the computer for the current management group connection.

Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

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

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

Specifies a user account under which the management group connection will run. It must have access to the server that is specified in the ComputerName parameter if specified. The default is the current user.

Enter a PSCredential object, such as one that is returned by the Get-Credential cmdlet.

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

## **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection.

Enter a Management group connection object, such as one returned by the Get-SCManagementGroupConnection cmdlet.

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

## **<CommonParameters>**

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

## **Examples**

### **----- EXAMPLE 1 -----**

This command gets the reporting server settings for the management group.

```
PS C:\>Get-SCOMReportingSetting
```

# Get-SCOMResourcePool

---

## Get-SCOMResourcePool

Gets resource pools.

### Syntax

Parameter Set: Empty

```
Get-SCOMResourcePool [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-EnableAutomaticMembership <Boolean> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromClassDisplayName

```
Get-SCOMResourcePool [-DisplayName] <String[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-EnableAutomaticMembership <Boolean> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromClassName

```
Get-SCOMResourcePool -Name <String[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-EnableAutomaticMembership <Boolean> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromId

```
Get-SCOMResourcePool -Id <Guid[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-EnableAutomaticMembership <Boolean> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromMember

```
Get-SCOMResourcePool [-Member] <ComputerHealthService[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-EnableAutomaticMembership <Boolean> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromObserver

```
Get-SCOMResourcePool [-Observer] <ComputerHealthService[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-EnableAutomaticMembership <Boolean> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCOMResourcePool cmdlet gets the specified resource pools.

### Parameters

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

Specifies the name of a computer to establish a connection with. The computer must be running the System Center Data Access Service. The default value is the computer for the current management group connection. Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

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

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

Specifies a user account under which the management group connection will run. The default is the current user. Type a user name, such as "User01", "Domain01\User01", or "User@Domain.com", or enter a PSCredential object, such as one returned by the Get-Credential cmdlet. When you type a user name, you will be prompted for a password. For more information, type Get-Help Get-Credential.

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

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

Specifies the display name of an object. This parameter takes one or more strings and will look for matches in the DisplayName property of the given objects that this cmdlet works with. DisplayName values may vary depending on which localized management packs are imported into the management group and the locale of the user running Windows PowerShell.

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

## **-EnableAutomaticMembership<Boolean>**

Returns only resource pools for which automatic membership is enabled. These resource pools contain only management servers.

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

## **-Id<Guid[]>**

Retrieves the resource pool with the specified GUID.

The GUID is stored in the Id property of the object that represents a resource pool. To get the GUID of a resource pool, type "Get-SCOMResourcePool | Format-Table DisplayName, Id".

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

## **-Member<ComputerHealthService[]>**

Specifies an object to include in the resource pool.

Valid objects that can be members of a resource pool include management servers and gateway servers. For information about how to get a management server object, type "Get-Help Get-SCOMManagementServer". For information about how to get a gateway server object, type "Get-Help Get-SCOMGatewayManagementServer".

Aliases	none
Required?	true
Position?	1
Default Value	none



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

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

Specifies the name of an object. This parameter takes one or more strings and looks for matches in the Name property of the given objects that this cmdlet works with.

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

### **-Observer<ComputerHealthService[]>**

Specifies a management server or a gateway management server that is not a member of the resource pool.

To make a resource pool highly available, you must add a minimum of three members to the pool, or two members and one observer.

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

### **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection. Specify a Management group connection object, such as one returned by the Get-SCOMManagementGroupConnection cmdlet.

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

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

## <CommonParameters>

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

## Examples

### ----- EXAMPLE 1 -----

Description

-----

This command gets the resource pools with "Management" in their display name and resource pool names that begin with "All".

```
PS C:\> Get-SCOMResourcePool -DisplayName *Management*, All*
```

### ----- EXAMPLE 2 -----

Description

-----

This command gets the resource pools with "02286611" in their name.

```
PS C:\> Get-SCOMResourcePool -Name *02286611*
```

### ----- EXAMPLE 3 -----

Description

-----

This command gets the resource pool with the Id of 7413b06b-a95b-4ae3-98f2-dac9ff76dabd of and the resource pool with the Id of 2ef74789-f9f5-46b0-af70-16d01d4f4577.

```
PS C:\> Get-SCOMResourcePool -Id 7413b06b-a95b-4ae3-98f2-dac9ff76dabd, 2ef74789-f9f5-46b0-af70-16d01d4f4577
```

## ----- EXAMPLE 4 -----

Description

-----

The first command gets the member with the display name Member01 and stores it in the \$Member variable.

The second command returns all resource pools for which Member01 (stored in the \$Member variable) is a member.

```
PS C:\> $Member = Get-SCOMManagementServer -Name "Member01"
```

```
PS C:\>Get-SCOMResourcePool -Member $Member
```

## ----- EXAMPLE 5 -----

Description

-----

The first command gets the observer with the display name Observer01 and stores it in the \$Observer variable.

The second command returns all resource pools for which Observer01 (stored in the \$Observer variable) is an observer.

```
PS C:\> $Observer = Get-SCOMManagementServer -Name "Observer01"
```

```
PS C:\>Get-SCOMResourcePool -Observer $Observer
```

### Related topics

[New-SCOMResourcePool](#)

[Remove-SCOMResourcePool](#)

[Set-SCOMResourcePool](#)

# Get-SCOMRMSEmulator

---

## Get-SCOMRMSEmulator

Gets the management server that hosts the RMS Emulator role.

### Syntax

Parameter Set: Empty

```
Get-SCOMRMSEmulator [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCOMRMSEmulator cmdlet gets the management server that currently hosts the RMS Emulator role.

### Parameters

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

Specifies the name of a computer to establish a connection with. The computer must be running the System Center Data Access Service. The default value is the computer for the current management group connection. Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

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

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

Specifies a user account under which the management group connection will run. The default is the current user. Type a user name, such as "User01", "Domain01\User01", or "User@Domain.com", or enter a PSCredential object, such as one returned by the Get-Credential cmdlet. When you type a user name, you will be prompted for a password. For more information, type Get-Help Get-Credential.

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

**-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection. Specify a Management group connection object, such as one returned by the Get-SCOMManagementGroupConnection cmdlet.

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

**<CommonParameters>**

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

**Examples**

----- **EXAMPLE 1** -----

Description

-----

This command establishes a temporary connection to Server01.Contoso.com and gets the management servers that holds the RMS Emulator role.

```
PS C:\>Get-SCMRMSEmulator -Computername "Server01.Contoso.com"
```

## Related topics

[Remove-SCOMRMSEmulator](#)

[Set-SCOMRMSEmulator](#)

# Get-SCOMRule

---

## Get-SCOMRule

Gets a list of monitoring rules.

### Syntax

Parameter Set: Empty

```
Get-SCOMRule [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromManagementPack

```
Get-SCOMRule [-ManagementPack <ManagementPack[]> ] [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromManagementPackClass

```
Get-SCOMRule [-Target] <ManagementPackClass[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromMonitoringDisplayRuleName

```
Get-SCOMRule [-DisplayName] <String[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromMonitoringRuleId

```
Get-SCOMRule -Id <Guid[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromMonitoringRuleName

```
Get-SCOMRule -Name <String[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

## Detailed Description

The Get-SCOMRule cmdlet gets a list of monitoring rules.

## Parameters

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

Specifies the name of a computer to establish a connection with. The computer must be running the System Center Data Access Service. The default value is the computer for the current management group connection. Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

Aliases	none
Required?	false

Position?	named
Default Value	localhost
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

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

Specifies a user account under which the management group connection will run. The default is the current user. Type a user name, such as "User01", "Domain01\User01", or "User@Domain.com", or enter a PSCredential object, such as one returned by the Get-Credential cmdlet. When you type a user name, you will be prompted for a password. For more information, type Get-Help Get-Credential.

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

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

Specifies the display name of an object. This parameter takes one or more strings and will look for matches in the DisplayName property of the given objects that this cmdlet works with. DisplayName values may vary depending on which localized management packs are imported into the management group and the locale of the user running Windows PowerShell.

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

### **-Id<Guid[]>**

Retrieves the monitoring rule with the specified GUID.



The GUID is stored in the Id property of the object that represents an SCGroup. To get the GUID of a class, type "Get-SCOMGroup | Format-Table DisplayName, Id".

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

### **-ManagementPack<ManagementPack[]>**

Retrieves monitoring rules for one or more specified management pack objects. Enter a variable that represents the management packs, or type a command that gets the management packs.

For information about how to get a management pack object, type Get-Help Get-SCOMManagementPack.

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

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

Specifies the name of an object.

This parameter takes one or more strings and will look for matches in the Name property of the given objects that this cmdlet works with.

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

## **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection. Specify a Management group connection object, such as one returned by the Get-SCOMManagementGroupConnection cmdlet.

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

## **-Target<ManagementPackClass[]>**

Retrieves monitoring rules for one or more specified target class objects. Enter a variable that represents the target classes, or type a command that gets the target classes.

For information about how to get a class object, type Get-Help Get-SCOMClass.

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

## **<CommonParameters>**

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

## **Examples**

### **----- EXAMPLE 1 -----**

Description

-----

This command gets all monitoring rules with "health" in their name.

```
PS C:\>Get-SCOMRule -Name *health*
```

## ----- EXAMPLE 2 -----

Description

-----

This command gets all monitoring rules with "performance" in their display name.

```
PS C:\>Get-SCOMRule -DisplayName *performance*
```

## ----- EXAMPLE 3 -----

Description

-----

This command gets all management packs with "System Center Core Monitoring" in their display name. It then uses the pipeline operator (|) to send the management pack objects to the Get-SCOMRule cmdlet which gets the monitoring rules for each management pack object.

```
PS C:\>Get-SCOMManagementPack -DisplayName "System Center Core Monitoring" | Get-SCOMRule
```

## ----- EXAMPLE 4 -----

Description

-----

This command gets the monitoring rules for each target class with "health" in their display name.

```
PS C:\>Get-SCOMRule -Target (Get-SCOMClass -DisplayName *health*)
```

## ----- EXAMPLE 5 -----

Description

-----

The first command gets the path to a management pack file and saves it in the \$MPFile variable.

The second command uses the ManagementPack parameter to get the monitoring rules contained in the management pack stored in the \$MPFile variable.

```
PS C:\>$MPFile = "D:\Program Files\System Center Operations Manager  
2007\Microsoft.SystemCenter.2007.mp"
```

```
PS C:\>Get-SCOMRule -ManagementPack $MPFile
```

## ----- EXAMPLE 6 -----

Description

-----

This example gets the monitoring rule with an Id of 7413b06b-a95b-4ae3-98f2-dac9ff76dabd.

```
PS C:\>Get-SCOMRule -Id 7413b06b-a95b-4ae3-98f2-dac9ff76dabd
```

## Related topics

[Get-SCOMClass](#)

[Get-SCOMManagementPack](#)

[Disable-SCOMRule](#)

[Enable-SCOMRule](#)

# Get-SCOMRunAsAccount

---

## Get-SCOMRunAsAccount

Gets Run As accounts.

### Syntax

Parameter Set: EmptyParameterSet

```
Get-SCOMRunAsAccount [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromId

```
Get-SCOMRunAsAccount -Id <Guid[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromRunAsAccountName

```
Get-SCOMRunAsAccount [-Name] <String[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCRunAsAccount cmdlet gets Run As accounts.

### Parameters

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

Specifies a computer to establish a connection with. The computer must be running the System Center Data Access service. The default is the computer for the current management group connection.

Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

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

## **-Credential<PSCredential>**

Specifies a user account under which the management group connection will run. It must have access to the server that is specified in the ComputerName parameter if specified. The default is the current user.

Enter a PSCredential object, such as one that is returned by the Get-Credential cmdlet.

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

## **-Id<Guid[]>**

The account ID.

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

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

The account name.

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

## **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection.

Enter a Management group connection object, such as one returned by the Get-SCManagementGroupConnection cmdlet.

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

## **<CommonParameters>**

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

## **Examples**

### **----- EXAMPLE 1 -----**

This command gets all action accounts.

```
PS C:\>Get-SCOMRunAsAccount
```

### **----- EXAMPLE 2 -----**

This command gets all data warehouse accounts.

```
PS C:\>Get-SCOMRunAsAccount -Name "Data Warehouse*"
```

# Get-SCOMRunAsDistribution

---

## Get-SCOMRunAsDistribution

Gets the distribution policy of an Operations Manager Run As account.

### Syntax

Parameter Set: Default

```
Get-SCOMRunAsDistribution [-RunAsAccount] <SecureData[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCOMRunAsDistribution cmdlet gets the distribution policy of an Operations Manager Run As account. Distribution policies determine which computers receive a runas account credential.

Due to the default formatting behavior of Windows PowerShell, it is possible that the console will not automatically display the list of approved distribution computers. To see the full list, save the output of this cmdlet to a variable, then inspect the .SecureDistribution parameter.

### Parameters

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

Specifies a computer to establish a connection with. The computer must be running the System Center Data Access service. The default is the computer for the current management group connection.

Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

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



## **-Credential<PSCredential>**

Specifies a user account under which the management group connection will run. It must have access to the server that is specified in the ComputerName parameter if specified. The default is the current user.

Enter a PSCredential object, such as one that is returned by the Get-Credential cmdlet.

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

## **-RunAsAccount<SecureData[]>**

The runas account. Enter a Microsoft.EnterpriseManagement.Security.SecureData object, such as one returned from Get-SCOMRunAsAccount.

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

## **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection.

Enter a Management group connection object, such as one returned by the Get-SCManagementGroupConnection cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## <CommonParameters>

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

## Examples

### ----- EXAMPLE 1 -----

This command displays the distribution policy for the contoso\administrator account.

```
PS C:\>Get-SCOMRunAsAccount "contoso\administrator" | Get-SCOMRunAsDistribution
```

### ----- EXAMPLE 2 -----

This example displays the distribution policy for the contoso\administrator account, and then lists the systems approved for distribution.

```
PS C:\>Get-SCOMRunAsAccount "contoso\administrator" | Get-SCOMRunAsDistribution | Tee-Object  
-Variable distribution
```

```
PS C:\>$distribution.SecureDistribution
```

# Get-SCOMRunAsProfile

---

## Get-SCOMRunAsProfile

Gets a list of Run As profiles.

### Syntax

Parameter Set: Empty

```
Get-SCOMRunAsProfile [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromId

```
Get-SCOMRunAsProfile -Id <Guid[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromManagementPack

```
Get-SCOMRunAsProfile [-ManagementPack <ManagementPack[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromRunAsProfileDisplayName

```
Get-SCOMRunAsProfile [-DisplayName <String[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromRunAsProfileName

```
Get-SCOMRunAsProfile -Name <String[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCOMRunAsProfile cmdlet gets a list of Run As profiles.

### Parameters

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

Specifies the name of a computer to establish a connection with. The computer must be running the System Center Data Access Service. The default value is the computer for the current management group connection. Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

Aliases	none
Required?	false
Position?	named
Default Value	none

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

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

Specifies a user account under which the management group connection will run. The default is the current user. Type a user name, such as "User01", "Domain01\User01", or "User@Domain.com", or enter a PSCredential object, such as one returned by the Get-Credential cmdlet. When you type a user name, you will be prompted for a password. For more information, type Get-Help Get-Credential.

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

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

Specifies the display name of an object. This parameter takes one or more strings and will look for matches in the DisplayName property of the given objects that this cmdlet works with. DisplayName values may vary depending on which localized management packs are imported into the management group and the locale of the user running Windows PowerShell.

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

### **-Id<Guid[]>**

Retrieves the Run As profile with the specified GUID.

An Id is a GUID that uniquely identifies an object on a local or remote computer. The GUID is unique, even when you have multiple sessions running in Windows PowerShell.

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

### **-ManagementPack<ManagementPack[]>**

Retrieves the Run As profiles for one or more management pack objects. Enter a variable that represents the management packs, or type a command that gets the management packs.

For information about how to get a management pack object, type Get-Help Get-SCOMManagementPack.

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

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

Specifies the name of an object.

This parameter takes one or more strings and will look for matches in the Name property of the given objects that this cmdlet works with.

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

## **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection. Specify a Management group connection object, such as one returned by the Get-SCOMManagementGroupConnection cmdlet.

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

## **<CommonParameters>**

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

## **Outputs**

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

- **Microsoft.EnterpriseManagement.Configuration.MonitoringSecureReference**

## **Examples**

### **----- EXAMPLE 1 -----**

Description

-----

This command gets the Run As profiles that have a name beginning with "Microsoft.SystemCenter".

```
PS C:\>Get-SCOMRunAsProfile -Name Microsoft.SystemCenter*
```

### **----- EXAMPLE 2 -----**

Description

-----

This command gets the Run As profile with the display name of "Automatic Agent Management Account".

```
PS C:\>Get-SCOMRunAsProfile -DisplayName "Automatic Agent Management Account"
```

### ----- EXAMPLE 3 -----

Description

-----

This command gets the Run As profile with an Id of 7413b06b-a95b-4ae3-98f2-dac9ff76dabd.

```
PS C:\>Get-SCOMRunAsProfile -Id 7413b06b-a95b-4ae3-98f2-dac9ff76dabd
```

### ----- EXAMPLE 4 -----

Description

-----

This command gets the management pack named "System Center Core Library" and uses the pipeline operator (|) to pass the management pack object to the Get-SCOMRunAsProfile cmdlet which gets all of the Run As profiles for the management pack object.

```
PS C:\>Get-SCOMManagementPack -displayname "System Center Core Library" | Get-SCOMRunAsProfile
```

## Related topics

[Get-SCOMManagementPack](#)

[Set-SCOMRunAsProfile](#)

# Get-SCOMTask

---

## Get-SCOMTask

Gets a list of tasks.

### Syntax

Parameter Set: Empty

```
Get-SCOMTask [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromInstance

```
Get-SCOMTask [-Instance] <EnterpriseManagementObject[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromManagementPack

```
Get-SCOMTask [-ManagementPack] <ManagementPack[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromManagementPackClass

```
Get-SCOMTask [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [-Target <ManagementPackClass[]> ] [ <CommonParameters>]
```

Parameter Set: FromTaskDisplayName

```
Get-SCOMTask [-DisplayName] <String[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromTaskId

```
Get-SCOMTask -Id <Guid[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromTaskName

```
Get-SCOMTask [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-Name <String[]> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

## Detailed Description

The Get-SCOMTask cmdlet gets a list of tasks.

## Parameters

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

Specifies the name of a computer to establish a connection with. The computer must be running the System Center Data Access Service. The default value is the computer for the current management group connection. Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).



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

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

Specifies a user account under which the management group connection will run. The default is the current user. Type a user name, such as "User01", "Domain01\User01", or "User@Domain.com", or enter a PSCredential object, such as one returned by the Get-Credential cmdlet. When you type a user name, you will be prompted for a password. For more information, type Get-Help Get-Credential.

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

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

Specifies the display name of an object. This parameter takes one or more strings and will look for matches in the DisplayName property of the given objects that this cmdlet works with. DisplayName values may vary depending on which localized management packs are imported into the management group and the locale of the user running Windows PowerShell.

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

## **-Id<Guid[]>**

Retrieves the task with the specified GUID.

The GUID is stored in the Id property of the object that represents an SCOMTask. To get the GUID of a task, type "Get-SCOMTask | Format-Table DisplayName, Id".

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

## **-Instance<EnterpriseManagementObject[]>**

Retrieves tasks for one or more class instance objects. Enter a variable that represents the class instances, or type a command that gets the class instances. This parameter also accepts group objects.

For information about how to get a class instance object, type Get-Help Get-SCOMClassInstance.

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

## **-ManagementPack<ManagementPack[]>**

Retrieves tasks for one or more management pack objects. Enter a variable that represents the management packs, or type a command that gets the management packs.

For information about how to get a management pack object, type Get-Help Get-SCOMManagementPack.

Aliases	none
Required?	true
Position?	1
Default Value	none

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

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

Specifies the name of an object.

This parameter takes one or more strings and will look for matches in the Name property of the given objects that this cmdlet works with.

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

### **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection. Specify a Management group connection object, such as one returned by the Get-SCOMManagementGroupConnection cmdlet.

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

### **-Target<ManagementPackClass[]>**

Retrieves tasks for one or more specified target class objects. Enter a variable that represents the target classes, or type a command that gets the target classes.

For information about how to get a class object, type Get-Help Get-SCOMClass.

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

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

## <CommonParameters>

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

## Examples

### ----- EXAMPLE 1 -----

Description

-----

This command gets all tasks with "health" in their name.

```
PS C:\>Get-SCOMTask -Name *health*
```

### ----- EXAMPLE 2 -----

Description

-----

This command gets all tasks that have a display name beginning with "Stop".

```
PS C:\>Get-SCOMTask -DisplayName "Stop*"
```

### ----- EXAMPLE 3 -----

Description

-----

This command gets all tasks for the management packs with "System Center Core" in their display name. By using the ErrorAction parameter with the SilentlyContinue value, if the command encounters a management pack with no associated tasks, the command will continue to run, and will not display errors.

```
PS C:\>Get-SCOMManagementPack -DisplayName "System Center Core*" | Get-SCOMTask -ErrorAction SilentlyContinue
```

## ----- EXAMPLE 4 -----

Description

-----

This command gets all tasks for classes with "health" in their name.

```
PS C:\>Get-SCOMClass -Name *health* | Get-SCOMTask
```

## ----- EXAMPLE 5 -----

Description

-----

This command gets all tasks for class instances that have "Contoso.com" in their name.

```
PS C:\>Get-SCOMClassInstance -Name *.Contoso.com | Get-SCOMTask
```

## ----- EXAMPLE 6 -----

Description

-----

This command gets the task with an Id of 7413b06b-a95b-4ae3-98f2-dac9ff76dabd.

```
PS C:\>Get-SCOMTask -Id 7413b06b-a95b-4ae3-98f2-dac9ff76dabd
```

## Related topics

[Get-SCOMClass](#)

[Get-SCOMClassInstance](#)

[Get-SCOMManagementPack](#)

[Get-SCOMTaskResult](#)

[Start-SCOMTask](#)

# Get-SCOMTaskResult

---

## Get-SCOMTaskResult

Gets the results for tasks that have run.

### Syntax

Parameter Set: Empty

```
Get-SCOMTaskResult [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromInstance

```
Get-SCOMTaskResult [-Instance] <EnterpriseManagementObject[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromTask

```
Get-SCOMTaskResult [-Task] <ManagementPackTask[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromTaskResultBatchId

```
Get-SCOMTaskResult [-BatchID] <Guid[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromTaskResultId

```
Get-SCOMTaskResult [-Id] <Guid[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCOMTaskResult cmdlet gets the results for tasks that have run.

### Parameters

#### **-BatchID<Guid[]>**

Retrieves task results for a batch in which a task was run.

A Batch Id is a GUID that uniquely identifies the batch in which a task was run.

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

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

Specifies the name of a computer to establish a connection with. The computer must be running the System Center Data Access Service. The default value is the computer for the current management group connection. Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

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

## **-Credential<PSCredential>**

Specifies a user account under which the management group connection will run. The default is the current user. Type a user name, such as "User01", "Domain01\User01", or "User@Domain.com", or enter a PSCredential object, such as one returned by the Get-Credential cmdlet. When you type a user name, you will be prompted for a password. For more information, type Get-Help Get-Credential.

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

## **-Id<Guid[]>**

Retrieves the task results with the specified Id.

An Id is a GUID that uniquely identifies an object on a local or remote computer. The GUID is unique, even when you have multiple sessions running in Windows PowerShell.

Aliases	none
Required?	true

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

### **-Instance<EnterpriseManagementObject[]>**

Retrieves task results for one or more specified class instance objects. Enter a variable that represents the class instances, or type a command that gets the class instances. This parameter also accepts group objects.

For information about how to get a class instance object, type Get-Help Get-SCOMClassInstance.

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

### **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection. Specify a Management group connection object, such as one returned by the Get-SCOMManagementGroupConnection cmdlet.

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

### **-Task<ManagementPackTask[]>**

Retrieves task results for one or more specified task objects. Enter a variable that represents the tasks, or type a command that gets the tasks.



For information about how to get a class instance object, type Get-Help Get-SCOMTask.

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

## <CommonParameters>

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

## Examples

### ----- EXAMPLE 1 -----

Description

-----

This command gets all tasks with "SystemCenter" in their name, and then returns the task results for those tasks.

By using the ErrorAction parameter with a value of SilentlyContinue, tasks that do not have matching task results will not show an error.

```
PS C:\>Get-SCOMTask -Name *SystemCenter* | -Get-SCOMTaskResult -ErrorAction SilentlyContinue
```

### ----- EXAMPLE 2 -----

Description

-----

This command gets all class instances in the Contoso.com domain and then returns the task results for each class instance object.

By using the ErrorAction parameter with a value of SilentlyContinue, class instances that do not have matching task results will not show an error.

```
PS C:\>Get-SCOMClassInstance -displayname "*.Consoso.com" | Get-SCOMTaskResult -ErrorAction SilentlyContinue
```

### ----- EXAMPLE 3 -----

Description

-----

This command gets the task results with an Id of 7413b06b-a95b-4ae3-98f2-dac9ff76dabd.

```
PS C:\>Get-SCOMTaskResult -Id 7413b06b-a95b-4ae3-98f2-dac9ff76dabd
```

#### ----- **EXAMPLE 4** -----

Description

-----

This command gets the task results with a BatchId of 2ef74789-f9f5-46b0-af70-16d01d4f4577.

```
PS C:\>Get-SCOMTaskResult -BatchId 2ef74789-f9f5-46b0-af70-16d01d4f4577
```

## **Related topics**

[Get-SCOMClassInstance](#)

[Get-SCOMTask](#)

[Start-SCOMTask](#)

# Get-SCOMTierConnector

---

## Get-SCOMTierConnector

Gets the connectors associated with a tiered management group.

### Syntax

Parameter Set: Empty

```
Get-SCOMTierConnector -Tier <TieredManagementGroup> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: DisplayName

```
Get-SCOMTierConnector [-DisplayName] <String[]> -Tier <TieredManagementGroup> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: Id

```
Get-SCOMTierConnector -Id <Guid[]> -Tier <TieredManagementGroup> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: Name

```
Get-SCOMTierConnector -Name <String[]> -Tier <TieredManagementGroup> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCOMTierConnector cmdlet gets the connectors associated with a tiered management group.

### Parameters

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

Specifies a computer to establish a connection with. The computer must be running the System Center Data Access service. The default is the computer for the current management group connection.

Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

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

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

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

Specifies a user account under which the management group connection will run. It must have access to the server that is specified in the ComputerName parameter if specified. The default is the current user.

Enter a PSCredential object, such as one that is returned by the Get-Credential cmdlet.

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

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

Specifies the display name of the connector.

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

### **-Id<Guid[]>**

Specifies the connector ID.

Aliases	none
Required?	true
Position?	named
Default Value	none

Accept Pipeline Input?	false
Accept Wildcard Characters?	false

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

Specifies the name of the connector.

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

### **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection.

Enter a Management group connection object, such as one returned by the Get-SCManagementGroupConnection cmdlet.

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

### **-Tier<TieredManagementGroup>**

Specifies the management group tier.

Aliases	none
Required?	true
Position?	named

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

## <CommonParameters>

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

## Examples

### ----- EXAMPLE 1 -----

This command gets all connectors for all tiered management groups.

```
PS C:\>Get-SCOMTieredManagementGroup | Get-SCOMTierConnector
```

### ----- EXAMPLE 2 -----

This command gets all connectors associated with the ContosoTier tiered management group, that have "myconnector" in their name.

```
PS C:\>Get-SCOMTieredManagementGroup "ContosoTier" | Get-SCOMTierConnector -Name  
"*myconnector*"
```

# Get-SCOMTieredManagementGroup

---

## Get-SCOMTieredManagementGroup

Gets tiered management groups defined in Operations Manager.

### Syntax

Parameter Set: Empty

```
Get-SCOMTieredManagementGroup [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-OnlyForConnector] [-SCSession <Connection[]> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: Id

```
Get-SCOMTieredManagementGroup -Id <Guid[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: Name

```
Get-SCOMTieredManagementGroup [-Name] <String[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The Get-SCOMTieredManagementGroup cmdlet gets tiered management groups defined in Operations Manager.

### Parameters

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

Specifies a computer to establish a connection with. The computer must be running the System Center Data Access service. The default is the computer for the current management group connection.

Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

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

## **-Credential<PSCredential>**

Specifies a user account under which the management group connection will run. It must have access to the server that is specified in the ComputerName parameter if specified. The default is the current user.

Enter a PSCredential object, such as one that is returned by the Get-Credential cmdlet.

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

## **-Id<Guid[]>**

The ID of the tiered management group.

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

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

The name of the tiered management group.

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



## **-OnlyForConnector**

If set, will only return tiered management groups which are available to connectors.

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

## **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection.

Enter a Management group connection object, such as one returned by the Get-SCManagementGroupConnection cmdlet.

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 -----

This command gets all tiered management groups.

```
PS C:\>Get-SCOMTieredManagementGroup
```

### ----- EXAMPLE 2 -----

This command gets the Fabrikam tiered management group.

```
PS C:\>Get-SCOMTieredManagementGroup -Name "Fabrikam"
```

### ----- EXAMPLE 3 -----

This command gets only those tiered management groups which are marked as available for connectors.

```
PS C:\>Get-SCOMTieredManagementGroup -OnlyForConnector
```

# Get-SCOMUserRole

---

## Get-SCOMUserRole

Gets user roles.

### Syntax

Parameter Set: Empty

```
Get-SCOMUserRole [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromUserRoleDisplayName

```
Get-SCOMUserRole [-DisplayName] <String[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromRoleId

```
Get-SCOMUserRole -Id <Guid[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromUserRoleName

```
Get-SCOMUserRole -Name <String[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCOMUserRole cmdlet gets user roles.

### Parameters

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

Specifies the name of a computer to establish a connection with. The computer must be running the System Center Data Access Service. The default value is the computer for the current management group connection. Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

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

## **-Credential<PSCredential>**

Specifies a user account under which the management group connection will run. The default is the current user.

Type a user name, such as "User01", "Domain01\User01", or "User@Domain.com", or enter a PSCredential object, such as one returned by the Get-Credential cmdlet. When you type a user name, you will be prompted for a password. For more information, type Get-Help Get-Credential.

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

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

Specifies the display name of an object. This parameter takes one or more strings and will look for matches in the DisplayName property of the given objects that this cmdlet works with. DisplayName values may vary depending on which localized management packs are imported into the management group and the locale of the user running Windows PowerShell.

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

## **-Id<Guid[]>**

Specifies the Guid of an object. This parameter takes one or more globally unique identifiers (Guids) and uses them to compare against the id property of the objects retrieved by this cmdlet.

Aliases	none
Required?	true
Position?	named

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

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

Specifies the name of an object. This parameter takes one or more strings and will look for matches in the Name property of the given objects that this cmdlet works with.

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

### **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection. Specify a Management group connection object, such as one returned by the Get-SCOMManagementGroupConnection cmdlet.

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

### **<CommonParameters>**

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

## Examples

### ----- EXAMPLE 1 -----

This command gets the user role information for the OperationsManagerReadOnlyOperators role.

```
PS C:\> Get-SCOMUserRole -Name OperationsManagerReadOnlyOperators
```

# Get-SCOMWebAddressSetting

---

## Get-SCOMWebAddressSetting

Gets the web console and online product knowledge addresses for the management group.

### Syntax

Parameter Set: Empty

```
Get-SCOMWebAddressSetting [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

### Detailed Description

The Get-SCOMWebAddressSetting cmdlet gets the web console and online product knowledge addresses for the management group.

### Parameters

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

Specifies a computer to establish a connection with. The computer must be running the System Center Data Access service. The default is the computer for the current management group connection.

Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

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

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

Specifies a user account under which the management group connection will run. It must have access to the server that is specified in the ComputerName parameter if specified. The default is the current user.

Enter a PSCredential object, such as one that is returned by the Get-Credential cmdlet.

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

### **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection.

Enter a Management group connection object, such as one returned by the Get-SCManagementGroupConnection cmdlet.

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

### **<CommonParameters>**

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

### **Examples**

#### **----- EXAMPLE 1 -----**

This command gets the web console and online product knowledge addresses for the management group.

```
PS C:\>Get-SCOMWebAddressSetting
```



# Import-SCOMManagementPack

---

## Import-SCOMManagementPack

Imports management packs.

### Syntax

Parameter Set: FromManagementPackFile

```
Import-SCOMManagementPack [-FullName] <String[]> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: \_\_AllParameterSets

```
Import-SCOMManagementPack [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-PassThru] [-SCSession <Connection[]> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: FromManagementPack

```
Import-SCOMManagementPack [-ManagementPack] <ManagementPack[]> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The Import-SCOMManagementPack cmdlet imports management packs.

Operations Manager attempts to validate the XML code of the management packs before the import. If the management pack contains XML code that is not valid, the management pack is not imported, and an error is returned.

### Parameters

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

Specifies a computer to establish a connection with. The computer must be running the System Center Data Access service. The default value is the computer for the current management group connection.

Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name (FQDN). To specify the local computer, type the computer name, "localhost", or a dot (.).

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

## **-Credential<PSCredential>**

Specifies a user account under which the management group connection will run. The account must have access to the server that is specified in the ComputerName parameter, if the server is specified. The default value is the current user.

You can enter a PSCredential object that is returned by the Get-Credential cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	User account of the current context.
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

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

Specifies the full paths and file names of the management packs to import. The list must contain files that are management packs (\*.xml, \*.mp) or management pack bundles (\*.mpb). If you specify a management pack bundle, all management packs from that bundle are imported.

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

## **-ManagementPack<ManagementPack[]>**

Specifies one or more management packs to import. You can enter a ManagementPack object that is returned by the Get-SCOMManagementPack cmdlet.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	true (ByValue)

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

## **-PassThru**

Passes the newly imported management pack to the pipeline. By default, this cmdlet does not generate any output.

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

## **-SCSession<Connection[]>**

Specifies a connection to a management server. The default value is the current management group connection.

You can enter a management group connection object that is returned by the Get-SCOMManagementGroupConnection cmdlet.

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](#)

## Inputs

The input type is the type of the objects that you can pipe to the cmdlet.

- **Microsoft.EnterpriseManagement.Configuration.ManagementPack** You can pipe a management pack to the ManagementPack parameter of the Import-SCOMManagementPack cmdlet.

## Outputs

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

- **Microsoft.EnterpriseManagement.Configuration.ManagementPack** An object that represents the management pack. It is available only when you use the -PassThru parameter.

## Notes

- If you specify multiple management packs, the order in which those management packs are imported will satisfy management pack interdependencies. As a result, management packs might be imported in an order that differs from the order that is provided at the command prompt.

## Examples

### ----- EXAMPLE 1 -----

This command imports the management pack file c:\temp\mymanagementpack.xml.

```
PS C:\>Import-SCOMManagementPack c:\temp\mymanagementpack.xml
```

### ----- EXAMPLE 2 -----

This command imports only the Woodgrove.ManagementPack management pack that is included in the c:\temp\mymanagementpackbundle.mpb management pack bundle file.

```
PS C:\>Import-SCOMManagementPack c:\temp\mymanagementpackbundle.mpb -name  
Woodgrove.ManagementPack
```

### ----- EXAMPLE 3 -----

This command imports only the Woodgrove.ManagementPack management pack that is included in the c:\temp\mymanagementpackbundle.mpb management pack bundle file, and displays information about the imported management pack.

```
PS C:\>Import-SCOMManagementPack c:\temp\mymanagementpackbundle.mpb -name  
Woodgrove.ManagementPack -passthru
```

## Related topics

[New-SCOMManagementPackBundle](#)

[Export-SCOMManagementPack](#)

[Get-SCOMManagementPack](#)

[New-SCOMManagementPack](#)

[Protect-SCOMManagementPack](#)

[Remove-SCOMManagementPack](#)

[Test-SCOMManagementPack](#)

# Install-SCOMAgent

---

## Install-SCOMAgent

Deploys one or more System Center Management agents using push install.

### Syntax

Parameter Set: Empty

```
Install-SCOMAgent -DNSHostName <String[]> -PrimaryManagementServer <ManagementServer> [-ActionAccount <PSCredential> ] [-AgentActionAccount <PSCredential> ] [-PassThru] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The Install-SCOMAgent cmdlet deploys one or more System Center Operations Manager agents using push install.

### Parameters

#### **-ActionAccount<PSCredential>**

Specifies the credentials under which the deployment task will run. If this parameter is omitted or has a null value, the default action account of the management server that will manage the agent is used.

This is not the action account that the agent will use after it is installed on the computer. By default, the action account the agent will use is set to Local System, but can be changed using the AgentActionAccount parameter.

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

#### **-AgentActionAccount<PSCredential>**

Specifies the action account that the agent will use after it is installed on a computer. If this parameter is omitted, the Local System account is used. Enter a variable that represents an action account, or type a

command that gets the action account. For information about how to get action accounts, type Get-Help Get-SCOMRunAsAccount.

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

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

Specifies the name of a Domain Name System (DNS) host.

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

### **-PassThru**

Returns an object representing the agent. By default, this cmdlet does not generate any output.

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

## **-PrimaryManagementServer<ManagementServer>**

Specifies the primary management server for the agent. Enter a variable that represents the management server, or type a command that gets the management server.

For information about how to get a management server object, type Get-Help Get-SCOMManagementServer.

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 -----

Description

-----

The first command gets the management server named "MgmtServer01.Contoso.com" and stores it in the \$PrimaryMgmtServer variable.

The second command installs an agent on Server01.Contoso.com, and sets its primary management server to the management server stored in the \$PrimaryMgmtServer variable.

```
PS C:\>$PrimaryMgmtServer = Get-SCOMManagementServer -Name "MgmtServer01.Contoso.com"
PS C:\>Install-SCOMAgent -Name "Server01.Contoso.com" -PrimaryManagementServer
$PrimaryMgmtServer
```

### ----- EXAMPLE 2 -----

Description

-----

The first command prompts the user for credentials and stores them in the \$InstallAccount variable.

The second command gets the management server named "MgmtServer01.Contoso.com" and stores it in the \$PrimaryMgmtServer variable.

The third command installs an agent on Server01.Contoso.com using the credentials supplied in the \$InstallAccount variable, and sets the primary management server for the agent to the management server stored in the \$PrimaryMgmtServer variable.

```
PS C:\>$InstallAccount = Get-Credential
PS C:\>$PrimaryMgmtServer = Get-SCOMManagementServer -Name "MgmtServer01.Contoso.com"
PS C:\>Install-SCOMAgent -Name "Server01.Contoso.com" -PrimaryManagementServer
$PrimaryMgmtServer -ActionAccount $InstallAccount
```

### ----- EXAMPLE 3 -----

Description

-----

The first command stores the name of three computers in the \$Agents variable.

The second command prompts the user for credentials and stores them in the \$InstallAccount variable.

The third command prompts the user for a password for a user account named ActionAccount in the Contoso domain and stores the credentials in the \$AgentActionAccount variable.

The fourth command gets the management server named "MgmtServer01.Contoso.com" and stores it in the \$PrimaryMgmtServer variable.

The fifth command installs an agent on the three computers using the credentials supplied in the \$InstallAccount variable, sets the primary management server for the agent to the management server stored in the \$PrimaryMgmtServer variable, and sets the agent action account to the ActionAccount user.

```
PS C:\>$Agents = "DC1.contoso.com","DC2.contoso.com","DC3.contoso.com"
```

```
PS C:\>$InstallAccount = Get-Credential
```

```
PS C:\>$AgentActionAccount = Get-Credential Contoso\ActionAccount
```

```
PS C:\>$PrimaryMgmtServer = Get-SCOMManagementserver -Name MgmtServer01.contoso.com
```

```
PS C:\>$Install-SCOMAgent -Name $Agents -PrimaryManagementServer $PrimaryMgmtServer -  
ActionAccount $InstallAccount -AgentActionAccount $AgentActionAccount -Confirm
```

## Related topics

[Get-SCOMRunAsAccount](#)

[Get-SCOMAgent](#)

[Get-SCOMManagementServer](#)

[Uninstall-SCOMAgent](#)

# Join-SCOMCEIP

---

## Join-SCOMCEIP

Enables Operations Manager CEIP data collection on the local computer.

### Syntax

Parameter Set: Empty

```
Join-SCOMCEIP [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession  
<Connection[]> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The Join-SCOMCEIP cmdlet enables Operations Manager Customer Experience Improvement Program (CEIP) data collection on the local computer. If enabled, it collects usage and performance data from the Operations Manager UI and cmdlets.

### Parameters

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

Specifies a computer to establish a connection with. The computer must be running the System Center Data Access service. The default is the computer for the current management group connection.

Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

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

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

Specifies a user account under which the management group connection will run. It must have access to the server that is specified in the ComputerName parameter if specified. The default is the current user.

Enter a PSCredential object, such as one that is returned by the Get-Credential cmdlet.

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

### **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection.

Enter a Management group connection object, such as one returned by the Get-SCManagementGroupConnection cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-Confirm**

Prompts you for confirmation before executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#)

## **Examples**

### **----- EXAMPLE 1 -----**

This command enables Operations Manager CEIP data collection on the local computer.

```
PS C:\>Join-SCOMCEIP
```

# New-SCOMLocation

---

## New-SCOMLocation

Creates a location to which you can assign agents.

### Syntax

Parameter Set: FromDisplayName

```
New-SCOMLocation [-DisplayName] <String> -Latitude <String> -Longitude <String> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [  
<CommonParameters>]
```

### Detailed Description

The New-SCOMLocation cmdlet creates a location to which you can assign agents. These locations display in the Web Application Availability Monitoring Summary Map Dashboard.

### Parameters

#### **-ComputerName<String[]>**

Specifies a computer to establish a connection with. The computer must be running the Operations Manager Data Access service.

Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

#### **-Credential<PSCredential>**

Specifies a user account under which the management group connection will run. The default is the current user. Type a user name, such as "User01", "Domain01\User01", or "User@Domain.com", or enter a PSCredential object, such as one returned by the Get-Credential cmdlet. When you type a user name, you will be prompted for a password. For more information, type Get-Help Get-Credential.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-DisplayName<String>**

Specifies the display name of the location.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-Latitude<String>**

Specifies the latitude for a location in decimal degrees.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-Longitude<String>**

Specifies the longitude for a location in decimal degrees.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection.

For information about how to get a management group connection, type `Get-Help Get-SCOMManagementGroupConnection` cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: `-Verbose`, `-Debug`, `-ErrorAction`, `-ErrorVariable`, `-OutBuffer`, and `-OutVariable`. For more information, see [about CommonParameters](#)

## **Examples**

### **----- EXAMPLE 1 -----**

Description

-----

This command creates a location with a display name of Seattle, WA, specifying the latitude and longitude of Seattle, and then displays information about the location to the user.

```
PS C:\> New-SCOMLocation -DisplayName "Seattle, WA" -Latitude 47.6063889 -Longitude -122.330833
```



## Related topics

[Get-SCOMLocation](#)

[Remove-SCOMLocation](#)

[Set-SCOMLocation](#)

[Update-SCOMLocation](#)

# New-SCOMManagementGroupConnection

---

## New-SCOMManagementGroupConnection

Creates a new connection for the specified management group.

### Syntax

Parameter Set: FromComputerNames

```
New-SCOMManagementGroupConnection [[-ComputerName] <String[]> ] [[-Credential] <PSCredential> ] [-PassThru] [ <CommonParameters>]
```

### Detailed Description

The New-SCOMManagementGroupConnection cmdlet creates a new connection for the specified management group. The last connection that is created with the New-SCOMManagementGroupConnection cmdlet becomes the active connection that is used by the Get- cmdlets when no –ComputerName or –SCSession parameter is specified. You can use the Set-SCOMManagementGroupConnection cmdlet to set a different active connection. If a connection already exists, this cmdlet does not create a new connection.

### Parameters

#### -ComputerName<String[]>

Specifies a computer to establish a connection with. The computer must be running the System Center Data Access service. The default value is the computer for the current management group connection.

Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name (FQDN). To specify the local computer, type the computer name, "localhost", or a dot (.).

Aliases	none
Required?	false
Position?	1
Default Value	localhost
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-Credential<PSCredential>**

Specifies a user account under which the management group connection will run. The account must have access to the server that is specified in the ComputerName parameter, if the server is specified. The default value is the current user.

You can enter a PSCredential object that is returned by the Get-Credential cmdlet.

Aliases	none
Required?	false
Position?	2
Default Value	Current user context
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-PassThru**

Specifies the output object that represents the new session. This output object can be passed to other cmdlets.

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](#)

## **Inputs**

The input type is the type of the objects that you can pipe to the cmdlet.

- **System.String** You can pipe a computer name to the ComputerName parameter of the New-SCOMManagementGroupConnection cmdlet.

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **Connection object.** This cmdlet can generate a connection object when it is used with the – PassThru parameter.

## Examples

### ----- EXAMPLE 1 -----

The commands in this example create a new management group connection.

```
PS C:\>New-SCOMManagementGroupConnection -ComputerName localhost -Credential (Get-Credential WOODGROVE\Administrator)
```

```
PS C:\>Get-SCOMManagementGroupConnection
```

## Related topics

[Get-SCOMManagementGroupConnection](#)

[Remove-SCOMManagementGroupConnection](#)

[Set-SCOMManagementGroupConnection](#)

# New-SCOMResourcePool

---

## New-SCOMResourcePool

Creates a resource pool.

### Syntax

Parameter Set: Empty

```
New-SCOMResourcePool [-DisplayName] <String> [-Member] <ComputerHealthService[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-Description <String> ] [-Observer <ComputerHealthService[]> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

### Detailed Description

The New-SCOMResourcePool cmdlet creates a resource pool. The new resource pool can contain management servers or gateway servers.

### Parameters

#### -ComputerName<String[]>

Specifies the name of a computer to establish a connection with. The computer must be running the System Center Data Access Service. The default value is the computer for the current management group connection. Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

#### -Credential<PSCredential>

Specifies a user account under which the management group connection will run. The default is the current user. Type a user name, such as "User01", "Domain01\User01", or "User@Domain.com", or enter a PSCredential object, such as one returned by the Get-Credential cmdlet. When you type a user name, you will be prompted for a password. For more information, type Get-Help Get-Credential.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-Description<String>**

Specifies a description for the resource pool. The maximum character length for this parameter is 4,000.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-DisplayName<String>**

Specifies the display name of an object. This parameter takes one or more strings and will look for matches in the DisplayName property of the given objects that this cmdlet works with. DisplayName values may vary depending on which localized management packs are imported into the management group and the locale of the user running Windows PowerShell.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-Member<ComputerHealthService[]>**

Specifies an object to include in the resource pool.

Valid objects that can be members of a resource pool include management servers or gateway servers. For information about how to get a management server object, type `Get-Help Get-SCOMManagementServer`. For information about how to get a gateway server object, type `Get-Help Get-SCOMGatewayManagementServer`.

Aliases	none
Required?	true
Position?	2
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-Observer<ComputerHealthService[]>**

Specifies a management server or a gateway management server that is not a member of the resource pool.

To make a resource pool highly available, you must add a minimum of three members to the pool, or two members and one observer.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection. Specify a Management group connection object, such as one returned by the `Get-SCOMManagementGroupConnection` cmdlet.

Aliases	none
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#)

## Examples

### ----- EXAMPLE 1 -----

Description

-----

This command creates a resource pool named Pool01 that contains all management servers.

The command in parentheses, which is executed first, gets all management servers. The cmdlet then passes the results of the command in parentheses to the New-SCOMResourcePool cmdlet. New-SCOMResourcePool then creates the resource pool and adds the management servers.

Note: Because the management servers are added to this resource pool manually, this resource pool will not have automatic membership enabled, and all future members will need to be added manually.

```
PS C:\> New-SCOMResourcePool -DisplayName "Pool01" -Member (Get-SCOMManagementServer) -
PassThru
```

## Related topics

[New-SCOMResourcePool](#)

[Remove-SCOMResourcePool](#)

[Set-SCOMResourcePool](#)



# Remove-SCOMADAgentAssignment

---

## Remove-SCOMADAgentAssignment

Deletes an Active Directory agent assignment from the management group.

### Syntax

Parameter Set: FromAgentAssignment

```
Remove-SCOMADAgentAssignment -AgentAssignment <AgentAssignment> -PrimaryServer  
<ManagementServer> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession  
<Connection[]> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The Remove-SCOMADAgentAssignment cmdlet deletes an Active Directory agent assignment from the management group.

### Parameters

#### **-AgentAssignment<AgentAssignment>**

The agent assignment to remove.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

#### **-ComputerName<String[]>**

Specifies a computer to establish a connection with. The computer must be running the System Center Data Access service. The default is the computer for the current management group connection.

Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-Credential<PSCredential>**

Specifies a user account under which the management group connection will run. It must have access to the server that is specified in the ComputerName parameter if specified. The default is the current user.

Enter a PSCredential object, such as one that is returned by the Get-Credential cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-PrimaryServer<ManagementServer>**

The primary management server used for the agent assignment being removed. This parameter is automatically populated when you pipe input from the Get-SCOMADAgentAssignment cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByPropertyName)
Accept Wildcard Characters?	false

## **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection.

Enter a Management group connection object, such as one returned by the Get-SCManagementGroupConnection cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-Confirm**

Prompts you for confirmation before executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#)

## Examples

### ----- EXAMPLE 1 -----

This command deletes all Active Directory agent assignments from server OMServer01.

```
PS C:\>Get-SCOMManagementServer "OMServer01*" | Get-SCOMADAgentAssignment | Remove-SCOMADAgentAssignment
```

### ----- EXAMPLE 2 -----

This command deletes all Active Directory agent assignments for the domain contoso.com.

```
PS C:\>Get-SCOMADAgentAssignment -Domain "contoso.com" | Remove-SCOMADAgentAssignment
```

# Remove-SCOMAgentlessManagedComputer

---

## Remove-SCOMAgentlessManagedComputer

Removes a computer that is not managed by an agent.

### Syntax

Parameter Set: Default

```
Remove-SCOMAgentlessManagedComputer [-Computer] <RemotelyManagedComputer[]> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The Remove-SCOMAgentlessManagedComputer cmdlet removes a computer that is not managed by an agent.

### Parameters

#### **-Computer<RemotelyManagedComputer[]>**

Specifies the agentless managed computer or computers to remove. Retrieve these objects using the Get-SCOMAgentlessManagedComputer cmdlet.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

#### **-Confirm**

Prompts you for confirmation before executing the command.

Required?	false
Position?	named

Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#)

## **Examples**

### **----- EXAMPLE 1 -----**

This example gets all computers with server01 in the name and then removes them from being managed agentlessly.

```
PS C:\> Get-SCOMAgentlesslyManagedComputer server01* | Remove-SCOMAgentlesslyManagedComputer
-Confirm
```

# Remove-SCOMAlertResolutionState

---

## Remove-SCOMAlertResolutionState

Removes a custom alert resolution state from the management group.

### Syntax

Parameter Set: FromManagementState

```
Remove-SCOMAlertResolutionState -ResolutionState <MonitoringAlertResolutionState[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The Remove-SCOMAlertResolutionState removes a custom alert resolution state from the management group.

### Parameters

#### **-ComputerName<String[]>**

Specifies a computer to establish a connection with. The computer must be running the System Center Data Access service. The default is the computer for the current management group connection.

Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

#### **-Credential<PSCredential>**

Specifies a user account under which the management group connection will run. It must have access to the server that is specified in the ComputerName parameter if specified. The default is the current user.

Enter a PSCredential object, such as one that is returned by the Get-Credential cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-ResolutionState<MonitoringAlertResolutionState[]>**

The custom resolution state to remove.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue, ByPropertyName)
Accept Wildcard Characters?	false

### **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection.

Enter a Management group connection object, such as one returned by the Get-SCManagementGroupConnection cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false



## -Confirm

Prompts you for confirmation before executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## -WhatIf

Describes what would happen if you executed the command without actually executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#)

## Examples

### ----- EXAMPLE 1 -----

This example adds a new custom alert resolution state named "Investigating", and then removes it.

```
PS C:\>Add-SCOMAlertResolutionState -Name Investigating -ResolutionStateCode 10
```

```
PS C:\>Get-SCOMAlertResolutionState -Name Investigating | Remove-SCOMAlertResolutionState
```

# Remove-SCOMConnector

---

## Remove-SCOMConnector

Removes a connector from the management group.

### Syntax

Parameter Set: FromConnector

```
Remove-SCOMConnector [-Connector] <MonitoringConnector[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

### Detailed Description

The Remove-SCOMConnector cmdlet removes a connector from the management group, disabling any transmission of information to that connector.

### Parameters

#### **-ComputerName<String[]>**

Specifies a computer to establish a connection with. The computer must be running the System Center Data Access service. The default is the computer for the current management group connection.

Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

#### **-Connector<MonitoringConnector[]>**

One or more monitoring connectors

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-Credential<PSCredential>**

Specifies a user account under which the management group connection will run. It must have access to the server that is specified in the ComputerName parameter if specified. The default is the current user.

Enter a PSCredential object, such as one that is returned by the Get-Credential cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection.

Enter a Management group connection object, such as one returned by the Get-SCManagementGroupConnection cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-Confirm**

Prompts you for confirmation before executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#)

## **Examples**

### **----- EXAMPLE 1 -----**

This command deletes the CustomConnector01 connector.

```
PS C:\>Get-SCOMConnector "CustomConnector01" | Remove-SCOMConnector
```

# Remove-SCOMDisabledClassInstance

---

## Remove-SCOMDisabledClassInstance

Deletes all class instances for which discovery has been disabled by using an override.

### Syntax

Parameter Set: Empty

```
Remove-SCOMDisabledClassInstance [-ComputerName <String[]> ] [-Credential <PSCredential> ]  
[-SCSession <Connection[]> ] [ <CommonParameters>]
```

### Detailed Description

The Remove-SCOMDisabledClassInstance cmdlet deletes all class instances for which discovery has been disabled by using an override. All the relationships that involve these class instances are also deleted.

Warning - This operation will result in class instances and relationships being permanently deleted. This operation is irreversible and will likely result in a significant amount of activity in the operational database.

### Parameters

#### -ComputerName<String[]>

Specifies the name of a computer to establish a connection with. The computer must be running the System Center Data Access Service. The default value is the computer for the current management group connection. Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-Credential<PSCredential>**

Specifies a user account under which the management group connection will run. The default is the current user. Type a user name, such as "User01", "Domain01\User01", or "User@Domain.com", or enter a PSCredential object, such as one returned by the Get-Credential cmdlet. When you type a user name, you will be prompted for a password. For more information, type Get-Help Get-Credential.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection. Specify a Management group connection object, such as one returned by the Get-SCOMManagementGroupConnection cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#)

## **Examples**

### **----- EXAMPLE 1 -----**

This command runs the Remove-SCOMDisabledClassInstance cmdlet interactively.

```
PS C:\> Remove-SCOMDisabledClassInstance
```

## ----- EXAMPLE 2 -----

This command runs the cmdlet as a job so that it runs asynchronously.

```
PS C:\> start-job -command "Remove-SCDisabledClassInstances"
```

# Remove-SCOMLocation

---

## Remove-SCOMLocation

Removes a location.

### Syntax

Parameter Set: FromAgent

```
Remove-SCOMLocation [-Agent] <AgentManagedComputer[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [-Confirm] [-WhatIf] [<CommonParameters>]
```

Parameter Set: FromLocation

```
Remove-SCOMLocation [-Location] <EnterpriseManagementObject[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [-Confirm] [-WhatIf] [<CommonParameters>]
```

Parameter Set: FromManagementServer

```
Remove-SCOMLocation [-ManagementServer] <ManagementServer[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [-Confirm] [-WhatIf] [<CommonParameters>]
```

Parameter Set: FromPool

```
Remove-SCOMLocation [-Pool] <ManagementServicePool[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [-Confirm] [-WhatIf] [<CommonParameters>]
```

### Detailed Description

The Remove-SCOMLocation cmdlet removes a location. For more information about locations, type: Get-Help New-SCOMLocation -detailed.

### Parameters

#### **-Agent<AgentManagedComputer[]>**

Specifies one or more agent objects. Enter a variable that represents the agents, or type a command that gets the agents.

For information about how to get agents, type Get-Help Get-SCOMAgent.

Aliases	none
Required?	true
Position?	1
Default Value	none



Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-ComputerName<String[]>**

Specifies a computer to establish a connection with. The computer must be running the Operations Manager Data Access service.

Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-Credential<PSCredential>**

Specifies a user account under which the management group connection will run. The default is the current user. Type a user name, such as "User01", "Domain01\User01", or "User@Domain.com", or enter a PSCredential object, such as one returned by the Get-Credential cmdlet. When you type a user name, you will be prompted for a password. For more information, type Get-Help Get-Credential.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-Location<EnterpriseManagementObject[]>**

Specifies a location object.

For information about how to get a location, type Get-Help Get-SCOMLocation.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-ManagementServer<ManagementServer[]>**

Specifies a primary management server object. For information about how to get a management server object, type Get-Help Get-SCOMManagementServer.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-Pool<ManagementServicePool[]>**

Specifies one or more resource pool objects. Enter a variable that represents the resource pools, or type a command that gets the resource pools.

For information about how to get a resource pool object, type "Get-Help Get-SCOMResourcePool".

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection.

For information about how to get a management group connection, type `Get-Help Get-SCOMManagementGroupConnection cmdlet`.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-Confirm**

Prompts you for confirmation before executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#)

## Examples

### ----- EXAMPLE 1 -----

Description

-----

The first command gets the location object with the display name of Seattle, WA and stores the object in the \$Location variable.

The second command deletes the location stored in \$Location, in this case, the location named Seattle, WA.

```
PS C:\> $Location = Get-SCOMLocation -DisplayName "Seattle, WA"
```

```
PS C:\> Remove-SCOMLocation -Location $Location
```

### ----- EXAMPLE 2 -----

Description

-----

The first command gets the agent object named Server01 and stores the object in the \$Agent variable.

The second command removes the location association from the agent-managed computer stored in \$Agent. This action does not delete the location itself.

```
PS C:\> $Agent = Get-SCOMAgent -Name "Server01.Contoso.com"
```

```
PS C:\> Remove-SCOMLocation -Agent $Agent
```

### ----- EXAMPLE 3 -----

Description

-----

The first command gets the management server object named MgmtServer01 and stores the object in the \$MgmtServer variable.

The second command removes the location association from the management server stored in \$MgmtServer. This action does not delete the location itself.

```
PS C:\> $MgmtServer = Get-SCOMManagementServer -Name "MgmtServer01.Contoso.com"
```

```
PS C:\> Remove-SCOMLocation -ManagementServer $MgmtServer
```

### ----- EXAMPLE 4 -----

Description

-----

The first command gets the resource pool object named Notifications Resource Pool and stores the object in the \$Pool variable.

The second command removes the location association from the resource pool stored in \$Pool. This action does not delete the location itself.

```
PS C:\> $Pool = Get-SCOMResourcePool -Name "Notifications Resource Pool"
```

```
PS C:\> Remove-SCOMLocation -Pool $Pool
```

## Related topics

[Get-SCOMLocation](#)

[New-SCOMLocation](#)

[Set-SCOMLocation](#)

# Remove-SCOMManagementGroupConnection

---

## Remove-SCOMManagementGroupConnection

Removes a management group connection.

### Syntax

Parameter Set: FromConnection

```
Remove-SCOMManagementGroupConnection [-Connection] <Connection[]> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

### Detailed Description

The Remove-SCOMManagementGroupConnection cmdlet removes a management group connection.

### Parameters

#### **-Connection<Connection[]>**

Specifies the connection, originating from the Get-SCOMManagementGroupConnection cmdlet, to be removed.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

#### **-Confirm**

Prompts you for confirmation before executing the command.

Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## -WhatIf

Describes what would happen if you executed the command without actually executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#)

## Inputs

The input type is the type of the objects that you can pipe to the cmdlet.

- **Microsoft.SystemCenter.Core.Connection.Connection** You can pipe a management group connection to the Connection parameter of the cmdlet.

## Outputs

The output type is the type of the objects that the cmdlet emits.

- This cmdlet does not generate any output.

## Examples

### ----- EXAMPLE 1 -----

The script in this example displays management server sessions (management group connections), removes the active management server session, and then displays available sessions to verify the removal.

```
PS C:\>Get-SCOMManagementGroupConnection
PS C:\>Get-SCOMManagementGroupConnection |?{$_.IsActive } | Remove-SCOMManagementGroupConnection
PS C:\>Get-SCOMManagementGroupConnection
```

## Related topics

[Get-SCOMManagementGroupConnection](#)

[New-SCOMManagementGroupConnection](#)

[Set-SCOMManagementGroupConnection](#)



# Remove-SCOMManagementPack

---

## Remove-SCOMManagementPack

Removes management packs.

### Syntax

Parameter Set: FromManagementPack

```
Remove-SCOMManagementPack [-ManagementPack] <ManagementPack[]> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

### Detailed Description

The Remove-SCOMManagementPack cmdlet removes management packs, along with all instances of types that are included in those management packs. You cannot remove a management pack on which other management packs, which are not being removed, are dependent.

### Parameters

#### **-ManagementPack<ManagementPack[]>**

Specifies the management pack to remove. You can specify a ManagementPack object that is returned by the Get-SCOMManagementPack cmdlet.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

#### **-Confirm**

Prompts you for confirmation before executing the command.

Required?	false
Position?	named

Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## -WhatIf

Describes what would happen if you executed the command without actually executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#)

## Inputs

The input type is the type of the objects that you can pipe to the cmdlet.

- **Microsoft.EnterpriseManagement.Configuration.ManagementPack** You can pipe a management pack to the ManagementPack parameter of the Remove-SCOMManagementPack cmdlet, for example, the object that is returned by the Get-SCOMManagementPack cmdlet.

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **None.** This cmdlet does not generate any output.

## Examples

### ----- EXAMPLE 1 -----

The pipelined command in this example removes all management packs in which the name matches the string "woodgrove".

```
PS C:\>Get-SCOMManagementPack | ?{ $_.name -match "Woodgrove" } | Remove-SCOMManagementPack
```

## ----- EXAMPLE 2 -----

The command in this example removes all management packs in which the name matches the string "Woodgrove".

```
PS C:\>$mps = Get-SCOMManagementPack | ?{ $_.name -match "Woodgrove" }; Remove-SCOMManagementPack $mps
```

### Related topics

[Import-SCOMManagementPack](#)

[New-SCOMManagementPackBundle](#)

[Export-SCOMManagementPack](#)

[Get-SCOMManagementPack](#)

[New-SCOMManagementPack](#)

[Protect-SCOMManagementPack](#)

[Test-SCOMManagementPack](#)

# Remove-SCOMNotificationChannel

---

## Remove-SCOMNotificationChannel

Removes a notification channel from the management group.

### Syntax

Parameter Set: Default

```
Remove-SCOMNotificationChannel [-Action] <NotificationAction> [[-Endpoint]  
<NotificationEndpoint> ] [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-  
SCSession <Connection[]> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The Remove-SCOMNotificationChannel cmdlet removes a notification channel from the management group.

### Parameters

#### **-Action<NotificationAction>**

The notification action used in the channel being removed. This parameter is automatically populated when you pipe input from Get-SCOMNotificationChannel.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	true (ByPropertyName)
Accept Wildcard Characters?	false

#### **-ComputerName<String[]>**

Specifies a computer to establish a connection with. The computer must be running the System Center Data Access service. The default is the computer for the current management group connection.

Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-Credential<PSCredential>**

Specifies a user account under which the management group connection will run. It must have access to the server that is specified in the ComputerName parameter if specified. The default is the current user.

Enter a PSCredential object, such as one that is returned by the Get-Credential cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-Endpoint<NotificationEndpoint>**

The notification endpoint used in the channel being removed. This parameter is automatically populated when you pipe input from Get-SCOMNotificationChannel.

Aliases	none
Required?	false
Position?	2
Default Value	none
Accept Pipeline Input?	true (ByPropertyName)
Accept Wildcard Characters?	false

## **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection.

Enter a Management group connection object, such as one returned by the Get-SCManagementGroupConnection cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-Confirm**

Prompts you for confirmation before executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#)

## Examples

### ----- EXAMPLE 1 -----

This command removes the EmailMeWhenAnythingHappens channel.

```
PS C:\>Get-SCOMNotificationChannel "EmailMeWhenAnythingHappens" | Remove-SCOMNotificationChannel
```

# Remove-SCOMNotificationSubscriber

---

## Remove-SCOMNotificationSubscriber

Removes a notification subscriber from the management group.

### Syntax

Parameter Set: FromNotificationRecipient

```
Remove-SCOMNotificationSubscriber [-Recipient] <NotificationRecipient[]> [-ComputerName  
<String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [-Confirm] [-WhatIf]  
[ <CommonParameters>]
```

### Detailed Description

The Remove-SCOMNotificationSubscriber cmdlet removes a notification subscriber from the management group.

### Parameters

#### **-ComputerName<String[]>**

Specifies a computer to establish a connection with. The computer must be running the System Center Data Access service. The default is the computer for the current management group connection.

Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

#### **-Credential<PSCredential>**

Specifies a user account under which the management group connection will run. It must have access to the server that is specified in the ComputerName parameter if specified. The default is the current user.



Enter a PSCredential object, such as one that is returned by the Get-Credential cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-Recipient<NotificationRecipient[]>**

The notification subscriber being removed.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	true (ByValue, ByPropertyName)
Accept Wildcard Characters?	false

### **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection.

Enter a Management group connection object, such as one returned by the Get-SCManagementGroupConnection cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-Confirm**

Prompts you for confirmation before executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#)

## **Examples**

### **----- EXAMPLE 1 -----**

This command removes the "Katarina" notification subscriber.

```
PS C:\>Get-SCOMNotificationSubscriber "Katarina" | Remove-SCOMNotificationSubscriber
```

# Remove-SCOMNotificationSubscription

---

## Remove-SCOMNotificationSubscription

Removes a notification subscription from the management group.

### Syntax

Parameter Set: FromNotificationSubscription

```
Remove-SCOMNotificationSubscription [-Subscription] <NotificationSubscription[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The Remove-SCOMNotificationSubscription cmdlet removes a notification subscription from the management group.

### Parameters

#### **-ComputerName<String[]>**

Specifies a computer to establish a connection with. The computer must be running the System Center Data Access service. The default is the computer for the current management group connection.

Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

#### **-Credential<PSCredential>**

Specifies a user account under which the management group connection will run. It must have access to the server that is specified in the ComputerName parameter if specified. The default is the current user.

Enter a PSCredential object, such as one that is returned by the Get-Credential cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection.

Enter a Management group connection object, such as one returned by the Get-SCManagementGroupConnection cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-Subscription<NotificationSubscription[]>**

The notification subscription to remove.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	true (ByValue, ByPropertyName)
Accept Wildcard Characters?	false

## -Confirm

Prompts you for confirmation before executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## -WhatIf

Describes what would happen if you executed the command without actually executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#)

## Examples

### ----- EXAMPLE 1 -----

This command removes the Subscription01 notification subscription.

```
PS C:\>Get-SCOMNotificationSubscription "Subscription01" | Remove-SCOMNotificationSubscription
```

# Remove-SCOMResourcePool

---

## Remove-SCOMResourcePool

Removes one or more resource pools.

### Syntax

Parameter Set: Empty

```
Remove-SCOMResourcePool [-ResourcePool] <ManagementServicePool[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The Remove-SCOMResourcePool cmdlet removes one or more resource pools.

This cmdlet requires a SCOMResourcePool object. For information about how to get a resource pool object, type "Get-Help Get-SCOMResourcePool".

### Parameters

#### **-ComputerName<String[]>**

Specifies the name of a computer to establish a connection with. The computer must be running the System Center Data Access Service. The default value is the computer for the current management group connection. Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

#### **-Credential<PSCredential>**

Specifies a user account under which the management group connection will run. The default is the current user. Type a user name, such as "User01", "Domain01\User01", or "User@Domain.com", or

enter a PSCredential object, such as one returned by the Get-Credential cmdlet. When you type a user name, you will be prompted for a password. For more information, type Get-Help Get-Credential.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-ResourcePool<ManagementServicePool[]>**

Specifies one or more resource pool objects. Enter a variable that represents the resource pools, or type a command that gets the resource pools.

For information about how to get a resource pool object, type "Get-Help Get-SCOMResourcePool".

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection. Specify a Management group connection object, such as one returned by the Get-SCOMManagementGroupConnection cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## -Confirm

Prompts you for confirmation before executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## -WhatIf

Describes what would happen if you executed the command without actually executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#)

## Examples

### ----- EXAMPLE 1 -----

Description

-----

This command gets all resource pools that have a display name beginning with "Pool01". It then prompts the user for confirmation before removing the resource pool.

```
PS C:\> Get-SCOMResourcePool -DisplayName Pool01* | Remove-SCOMResourcePool -Confirm
```

## Related topics

[Get-SCResourcePool](#)



[New-SCResourcePool](#)

[Set-SCResourcePool](#)

# Remove-SCOMRMSEmulator

---

## Remove-SCOMRMSEmulator

Removes the RMS Emulator Role.

### Syntax

Parameter Set: Default

```
Remove-SCOMRMSEmulator [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The Remove-SCOMRMSEmulator cmdlet removes the RMS Emulator Role from the management group. It will prompt you to validate that you actually want to perform this action.

### Parameters

#### **-ComputerName<String[]>**

Specifies the name of a computer to establish a connection with. The computer must be running the System Center Data Access Service. The default value is the computer for the current management group connection. Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

Aliases	none
Required?	false
Position?	named
Default Value	localhost
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

#### **-Credential<PSCredential>**

Specifies a user account under which the management group connection will run. The default is the current user. Type a user name, such as "User01", "Domain01\User01", or "User@Domain.com", or enter a PSCredential object, such as one returned by the Get-Credential cmdlet. When you type a user name, you will be prompted for a password. For more information, type Get-Help Get-Credential.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection. Specify a Management group connection object, such as one returned by the Get-SCOMManagementGroupConnection cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-Confirm**

Prompts you for confirmation before executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#)

## Examples

### ----- EXAMPLE 1 -----

Description

-----

This command removes the RMS Emulator role from Server01, prompting for confirmation prior to performing the operation.

```
PS C:\>Remove-SCOMRMSEmulator -ComputerName "Server01.Contoso.com" -Confirm
```

## Related topics

[Get-SCOMRMSEmulator](#)

[Remove-SCOMRMSEmulator](#)

# Remove-SCOMRunAsAccount

---

## Remove-SCOMRunAsAccount

Removes a Run As account from the management group.

### Syntax

Parameter Set: Default

```
Remove-SCOMRunAsAccount [-RunAsAccount] <SecureData[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

### Detailed Description

The Remove-SCOMRunAsAccount cmdlet removes a Run As account from the management group. The account must not be part of any existing Run As profiles, or the command will fail.

### Parameters

#### **-ComputerName<String[]>**

Specifies a computer to establish a connection with. The computer must be running the System Center Data Access service. The default is the computer for the current management group connection.

Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

#### **-Credential<PSCredential>**

Specifies a user account under which the management group connection will run. It must have access to the server that is specified in the ComputerName parameter if specified. The default is the current user.

Enter a PSCredential object, such as one that is returned by the Get-Credential cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-RunAsAccount<SecureData[]>**

The Run As account to remove. Ensure this account is not part of any Run As profiles.

Enter a Microsoft.EnterpriseManagement.Security.SecureData object, such as one returned from Get-SCOMRunAsAccount.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection.

Enter a Management group connection object, such as one returned by the Get-SCManagementGroupConnection cmdlet.

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 -----

This command removes the contoso\administrator Run As account.

```
PS C:\>Get-SCOMRunAsAccount -Name 'contoso\administrator' | Remove-SCOMRunAsAccount
```

# Remove-SCOMRunAsProfile

---

## Remove-SCOMRunAsProfile

Removes a Run as profile from the management group.

### Syntax

Parameter Set: FromRunAsProfile

```
Remove-SCOMRunAsProfile [-RunAsProfile] <ManagementPackSecureReference[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The Remove-SCOMRunAsProfile cmdlet removes a Run as profile from the management group.

### Parameters

#### **-ComputerName<String[]>**

Specifies a computer to establish a connection with. The computer must be running the System Center Data Access service. The default is the computer for the current management group connection.

Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

#### **-Credential<PSCredential>**

Specifies a user account under which the management group connection will run. It must have access to the server that is specified in the ComputerName parameter if specified. The default is the current user.

Enter a PSCredential object, such as one that is returned by the Get-Credential cmdlet.



Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-RunAsProfile<ManagementPackSecureReference[]>**

The runas profile to remove.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection.

Enter a Management group connection object, such as one returned by the Get-SCManagementGroupConnection cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-Confirm**

Prompts you for confirmation before executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#)

## **Examples**

### **----- EXAMPLE 1 -----**

This command deletes the Profile01 profile.

```
PS C:\>Get-SCOMRunAsProfile "Profile01" | Remove-SCOMRunAsProfile
```

# Remove-SCOMSubscriberSchedule

---

## Remove-SCOMSubscriberSchedule

Removes an entry from a notification subscriber schedule.

### Syntax

Parameter Set: Default

```
Remove-SCOMSubscriberSchedule [-Subscriber] <NotificationRecipient> [-Entry]  
<NotificationRecipientScheduleEntry> [-ComputerName <String[]> ] [-Credential <PSCredential>  
] [-PassThru] [-SCSession <Connection[]> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The Remove-SCOMSubscriberSchedule cmdlet removes an entry from a notification subscriber schedule.

### Parameters

#### **-ComputerName<String[]>**

Specifies a computer to establish a connection with. The computer must be running the System Center Data Access service. The default is the computer for the current management group connection.

Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

#### **-Credential<PSCredential>**

Specifies a user account under which the management group connection will run. It must have access to the server that is specified in the ComputerName parameter if specified. The default is the current user.

Enter a PSCredential object, such as one that is returned by the Get-Credential cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-Entry<NotificationRecipientScheduleEntry>**

The schedule entry which will be removed.

Aliases	none
Required?	true
Position?	2
Default Value	none
Accept Pipeline Input?	true (ByValue, ByPropertyName)
Accept Wildcard Characters?	false

### **-PassThru**

Specifies a connection to a management server. The default is the current management group connection.

Enter a Management group connection object, such as one returned by the Get-SCManagementGroupConnection cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection.

Enter a Management group connection object, such as one returned by the Get-SCManagementGroupConnection cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-Subscriber<NotificationRecipient>**

The notification subscriber whose schedule entries will be removed.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	true (ByValue, ByPropertyName)
Accept Wildcard Characters?	false

## **-Confirm**

Prompts you for confirmation before executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## -WhatIf

Describes what would happen if you executed the command without actually executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#)

## Examples

### ----- EXAMPLE 1 -----

This example removes the last schedule entry from a subscriber.

```
PS C:\>$subscriber = Get-SCOMNotificationSubscriber "Katarina"
```

```
PS C:\>$subscriber.ScheduleEntries | Select-Object -Last 1 | Remove-ScomSubscriberSchedule -Subscriber $subscriber
```

# Remove-SCOMTierConnector

---

## Remove-SCOMTierConnector

Disassociates a connector from a tiered management group.

### Syntax

Parameter Set: Default

```
Remove-SCOMTierConnector -Connector <MonitoringConnector> -Tier <TieredManagementGroup> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The Remove-SCOMTierConnector cmdlet disassociates a connector from a tiered management group. Removing a connector from a tier stops that tier from transmitting monitoring information to that connector.

### Parameters

#### **-ComputerName<String[]>**

Specifies a computer to establish a connection with. The computer must be running the System Center Data Access service. The default is the computer for the current management group connection.

Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

#### **-Connector<MonitoringConnector>**

Specifies the connector to remove.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-Credential<PSCredential>**

Specifies a user account under which the management group connection will run. It must have access to the server that is specified in the ComputerName parameter if specified. The default is the current user.

Enter a PSCredential object, such as one that is returned by the Get-Credential cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection.

Enter a Management group connection object, such as one returned by the Get-SCManagementGroupConnection cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false



## **-Tier<TieredManagementGroup>**

Specifies the tiered management group from which to remove the connector.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-Confirm**

Prompts you for confirmation before executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#)

## Examples

### ----- EXAMPLE 1 -----

This example clears all connectors from the first defined tiered management group.

```
PS C:\>$tier = Get-SCOMTieredManagementGroup | Select-Object -First 1
```

```
PS C:\>Get-SCOMTierConnector -Tier $tier | Remove-SCOMTierConnector -Tier $tier
```

# Remove-SCOMTieredManagementGroup

---

## Remove-SCOMTieredManagementGroup

Removes a tiered management group.

### Syntax

Parameter Set: Default

```
Remove-SCOMTieredManagementGroup -Tier <TieredManagementGroup[]> [-ComputerName <String[]> ]  
[-Credential <PSCredential> ] [-SCSession <Connection[]> ] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

### Detailed Description

The Remove-SCOMTieredManagementGroup cmdlet removes a tiered management group.

### Parameters

#### **-ComputerName<String[]>**

Specifies a computer to establish a connection with. The computer must be running the System Center Data Access service. The default is the computer for the current management group connection.

Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

#### **-Credential<PSCredential>**

Specifies a user account under which the management group connection will run. It must have access to the server that is specified in the ComputerName parameter if specified. The default is the current user.

Enter a PSCredential object, such as one that is returned by the Get-Credential cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection.

Enter a Management group connection object, such as one returned by the Get-SCManagementGroupConnection cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Tier<TieredManagementGroup[]>**

Specifies a connection to a management server. The default is the current management group connection.

Enter a Management group connection object, such as one returned by the Get-SCManagementGroupConnection 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](#)

## **Examples**

### **----- EXAMPLE 1 -----**

This command removes all tiered management groups.

```
PS C:\>Get-SCOMTieredManagementGroup | Remove-SCOMTieredManagementGroup
```

# Repair-SCOMAgent

---

## Repair-SCOMAgent

Repairs an Operations Manager agent.

### Syntax

Parameter Set: FromAgent

```
Repair-SCOMAgent -Agent <AgentManagedComputer[]> [-Actionaccount <PSCredential> ] [-PassThru] [ <CommonParameters>]
```

### Detailed Description

The Repair-SCOMAgent cmdlet repairs the specified Operations Manager agent.

### Parameters

#### **-Actionaccount<PSCredential>**

Specifies the credentials under which the task will run. If this parameter is omitted or has a null value, the default action account of the management server that manages the agent is used.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

#### **-Agent<AgentManagedComputer[]>**

Specifies one or more agent objects to repair. Enter a variable that represents the agents, or type a command that gets the agents. For information about how to get agents, type Get-Help Get-SCOMAgent.

Aliases	none
Required?	true

Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-PassThru**

Returns an object representing the agent. By default, this cmdlet does not generate any output.

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 -----**

This example gets the Operations Manager agent named Server01.Contoso.Com and uses the pipeline operator to pass the agent to the Repair-SCOMAgent cmdlet.

```
PS C:\> Get-SCOMAgent -Name "Server01.Contoso.Com" | Repair-SCOMAgent
```

# Resolve-SCOMAlert

---

## Resolve-SCOMAlert

Resolves an alert. This does the same action as Set-SCOMAlert -ResolutionState 255.

### Syntax

Parameter Set: FromAlertDefault

```
Resolve-SCOMAlert [[-Comment] <String> ] [[-CustomField1] <String> ] [[-CustomField2] <String> ] [[-CustomField3] <String> ] [[-CustomField4] <String> ] [[-CustomField5] <String> ] [[-CustomField6] <String> ] [[-CustomField7] <String> ] [[-CustomField8] <String> ] [[-CustomField9] <String> ] [[-CustomField10] <String> ] [[-Owner] <String> ] [[-TicketId] <String> ] -Alert <MonitoringAlert[]> [-Connector <MonitoringConnector> ] [-PassThru] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The Resolve-SCOMAlert cmdlet resolves an alert in Operations Manager by setting the resolutionstate on the alert to Closed (255).

### Parameters

#### **-Alert<MonitoringAlert[]>**

The monitoring alert that will be resolved.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue, ByPropertyName)
Accept Wildcard Characters?	false

#### **-Comment<String>**

A comment to set when resolving the alert.

Aliases	none
---------	------



Required?	false
Position?	1
Default Value	none
Accept Pipeline Input?	true (ByValue, ByPropertyName)
Accept Wildcard Characters?	false

**-Connector<MonitoringConnector>**

The monitoring connector.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue, ByPropertyName)
Accept Wildcard Characters?	false

**-CustomField1<String>**

Any information to place in CustomField1.

Aliases	none
Required?	false
Position?	2
Default Value	none
Accept Pipeline Input?	true (ByValue, ByPropertyName)
Accept Wildcard Characters?	false

**-CustomField10<String>**

Any information to place in CustomField10.

Aliases	none
Required?	false

Position?	11
Default Value	none
Accept Pipeline Input?	true (ByValue, ByPropertyName)
Accept Wildcard Characters?	false

### **-CustomField2<String>**

Any information to place in CustomField2.

Aliases	none
Required?	false
Position?	3
Default Value	none
Accept Pipeline Input?	true (ByValue, ByPropertyName)
Accept Wildcard Characters?	false

### **-CustomField3<String>**

Any information to place in CustomField3.

Aliases	none
Required?	false
Position?	4
Default Value	none
Accept Pipeline Input?	true (ByValue, ByPropertyName)
Accept Wildcard Characters?	false

### **-CustomField4<String>**

Any information to place in CustomField4.

Aliases	none
Required?	false
Position?	5

Default Value	none
Accept Pipeline Input?	true (ByValue, ByPropertyName)
Accept Wildcard Characters?	false

### **-CustomField5<String>**

Any information to place in CustomField5.

Aliases	none
Required?	false
Position?	6
Default Value	none
Accept Pipeline Input?	true (ByValue, ByPropertyName)
Accept Wildcard Characters?	false

### **-CustomField6<String>**

Any information to place in CustomField6.

Aliases	none
Required?	false
Position?	7
Default Value	none
Accept Pipeline Input?	true (ByValue, ByPropertyName)
Accept Wildcard Characters?	false

### **-CustomField7<String>**

Any information to place in CustomField7.

Aliases	none
Required?	false
Position?	8
Default Value	none

Accept Pipeline Input?	true (ByValue, ByPropertyName)
Accept Wildcard Characters?	false

### **-CustomField8<String>**

Any information to place in CustomField8.

Aliases	none
Required?	false
Position?	9
Default Value	none
Accept Pipeline Input?	true (ByValue, ByPropertyName)
Accept Wildcard Characters?	false

### **-CustomField9<String>**

Any information to place in CustomField9.

Aliases	none
Required?	false
Position?	10
Default Value	none
Accept Pipeline Input?	true (ByValue, ByPropertyName)
Accept Wildcard Characters?	false

### **-Owner<String>**

The owner of the alert.

Aliases	none
Required?	false
Position?	13
Default Value	none
Accept Pipeline Input?	true (ByValue, ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

## **-PassThru**

If set, returns the resolved alert.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByPropertyName)
Accept Wildcard Characters?	false

## **-TicketId<String>**

The ticket ID.

Aliases	none
Required?	false
Position?	14
Default Value	none
Accept Pipeline Input?	true (ByValue, ByPropertyName)
Accept Wildcard Characters?	false

## **-Confirm**

Prompts you for confirmation before executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#)

## **Examples**

### **----- EXAMPLE 1 -----**

This command resolves all Error alerts.

```
PS C:\>Get-SCOMAlert -Severity 2 | Resolve-SCOMAlert -Comment 'All alerts are resolved.'
```

# Set-SCOMAgentApprovalSetting

---

## Set-SCOMAgentApprovalSetting

Sets the manual agent approval setting for the management group.

### Syntax

Parameter Set: AutoReject

```
Set-SCOMAgentApprovalSetting -AutoReject[-ComputerName <String[]> ] [-Credential <PSCredential> ] [-PassThru] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: AutoApprove

```
Set-SCOMAgentApprovalSetting -AutoApprove[-ComputerName <String[]> ] [-Credential <PSCredential> ] [-PassThru] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: Pending

```
Set-SCOMAgentApprovalSetting -Pending[-ComputerName <String[]> ] [-Credential <PSCredential> ] [-PassThru] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

### Detailed Description

The Set-SCOMAgentApprovalSetting cmdlet sets the manual agent approval setting for the management group. New manual agents can be automatically approved (AutoApprove), automatically rejected (AutoReject), or set to Pending.

### Parameters

#### -AutoApprove

If specified, any new manual agents contacting a management server will automatically be approved and joined to the management group.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## -AutoReject

If specified, any new manual agents contacting a management server will automatically be rejected and not allowed to join the management group.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## -ComputerName<String[]>

Specifies a computer to establish a connection with. The computer must be running the System Center Data Access service. The default is the computer for the current management group connection.

Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## -Credential<PSCredential>

Specifies a user account under which the management group connection will run. It must have access to the server that is specified in the ComputerName parameter if specified. The default is the current user.

Enter a PSCredential object, such as one that is returned by the Get-Credential cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none



Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-PassThru**

If specified, the setting will be returned to the pipeline.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Pending**

If specified, any new manual agents contacting a management server will be placed in the pending management list. Use the Get-SCOMPendingManagement, Approve-SCOMPendingManagement, and Deny-SCOMPendingManagement cmdlets to manage agents in the pending management list.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection.

Enter a Management group connection object, such as one returned by the Get-SCManagementGroupConnection cmdlet.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#)

## Examples

### ----- EXAMPLE 1 -----

This command sets the manual agent approval setting for the management group to AutoApprove.

```
PS C:\>Set-SCOMAgentApprovalSetting -AutoApprove
```

### ----- EXAMPLE 2 -----

This command sets the manual agent approval setting for the management group to AutoReject.

```
PS C:\>Set-SCOMAgentApprovalSetting -AutoReject
```

### ----- EXAMPLE 3 -----

This command sets the manual agent approval setting for the management group to set all new manual requests to pending.

```
PS C:\>Set-SCOMAgentApprovalSetting -Pending
```

# Set-SCOMAgentlessManagedComputer

---

## Set-SCOMAgentlessManagedComputer

Sets the configuration for managed computers that do not have agents.

### Syntax

Parameter Set: FromAgentManagedBy

```
Set-SCOMAgentlessManagedComputer [-Computer] <RemotelyManagedComputer[]> [-ManagedByAgent] <AgentManagedComputer> [-PassThru] [ <CommonParameters>]
```

Parameter Set: FromManagementServerManagedBy

```
Set-SCOMAgentlessManagedComputer [-Computer] <RemotelyManagedComputer[]> [-ManagedByManagementServer] <ManagementServer> [-PassThru] [ <CommonParameters>]
```

### Detailed Description

The Set-SCOMAgentlessManagedComputer cmdlet sets the configuration for managed computers that do not have agents.

### Parameters

#### **-Computer<RemotelyManagedComputer[]>**

Specifies the computer or computers which are being managed without agents.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

#### **-ManagedByAgent<AgentManagedComputer>**

Specifies the agent that will perform the agentless management.

Aliases	none
---------	------

Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-ManagedByManagementServer<ManagementServer>**

Specifies the management server that will perform the agentless management.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-PassThru**

Passes the object created by this cmdlet through the pipeline. By default, this cmdlet does not pass any objects through the pipeline.

This parameter returns an object representing the added content. By default, this cmdlet does not generate any output.

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 -----

This command sets OMAgent01 as the agent that will perform agentless management for Server02.

```
PS C:\> Get-SCOMAgentlessManagedComputer -Name "Server02.Contoso.com" | Set-SCOMAgentlessManagedComputer -ManagedByAgent (Get-SCOMAgent OMAgent01.Contoso.com) -PassThru
```

# Set-SCOMAlert

---

## Set-SCOMAlert

Changes the properties of the specified alert.

### Syntax

Parameter Set: FromAlertDefault

```
Set-SCOMAlert [[-Comment] <String> ] [[-CustomField1] <String> ] [[-CustomField2] <String> ]  
[[ -CustomField3] <String> ] [[-CustomField4] <String> ] [[-CustomField5] <String> ] [[-  
CustomField6] <String> ] [[-CustomField7] <String> ] [[-CustomField8] <String> ] [[-  
CustomField9] <String> ] [[-CustomField10] <String> ] [[-ResolutionState] <Byte> ] [[-Owner]  
<String> ] [[-TicketId] <String> ] -Alert <MonitoringAlert[]> [-Connector  
<MonitoringConnector> ] [-PassThru] [ <CommonParameters>]
```

### Detailed Description

The Set-SCOMAlert cmdlet changes the properties of the specified alert.

### Parameters

#### **-Alert<MonitoringAlert[]>**

Specifies one or more alert objects. Enter a variable that represents the alerts, or type a command that gets the alerts. For information about how to get an alert object, type Get-Help Get-SCOMAlert.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue, ByPropertyName)
Accept Wildcard Characters?	false

#### **-Comment<String>**

Specifies a comment.

Aliases	none
---------	------

Required?	false
Position?	1
Default Value	none
Accept Pipeline Input?	true (ByValue, ByPropertyName)
Accept Wildcard Characters?	false

### **-Connector<MonitoringConnector>**

Specifies a monitoring connector object. Enter a variable that represents the connector, or type a command that gets the connector.

For information about how to get a connector object, type Get-Help Get-SCOMConnector.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue, ByPropertyName)
Accept Wildcard Characters?	false

### **-CustomField1<String>**

Specifies a custom field property on an alert object.

Aliases	none
Required?	false
Position?	2
Default Value	none
Accept Pipeline Input?	true (ByValue, ByPropertyName)
Accept Wildcard Characters?	false

### **-CustomField10<String>**

Specifies a custom field property on an alert object.

Aliases	none
Required?	false
Position?	11
Default Value	none
Accept Pipeline Input?	true (ByValue, ByPropertyName)
Accept Wildcard Characters?	false

### **-CustomField2<String>**

Specifies a custom field property on an alert object.

Aliases	none
Required?	false
Position?	3
Default Value	none
Accept Pipeline Input?	true (ByValue, ByPropertyName)
Accept Wildcard Characters?	false

### **-CustomField3<String>**

Specifies a custom field property on an alert object.

Aliases	none
Required?	false
Position?	4
Default Value	none
Accept Pipeline Input?	true (ByValue, ByPropertyName)
Accept Wildcard Characters?	false

### **-CustomField4<String>**

Specifies a custom field property on an alert object.

Aliases	none
---------	------



Required?	false
Position?	5
Default Value	none
Accept Pipeline Input?	true (ByValue, ByPropertyName)
Accept Wildcard Characters?	false

### **-CustomField5<String>**

Specifies a custom field property on an alert object.

Aliases	none
Required?	false
Position?	6
Default Value	none
Accept Pipeline Input?	true (ByValue, ByPropertyName)
Accept Wildcard Characters?	false

### **-CustomField6<String>**

Specifies a custom field property on an alert object.

Aliases	none
Required?	false
Position?	7
Default Value	none
Accept Pipeline Input?	true (ByValue, ByPropertyName)
Accept Wildcard Characters?	false

### **-CustomField7<String>**

Specifies a custom field property on an alert object.

Aliases	none
Required?	false

Position?	8
Default Value	none
Accept Pipeline Input?	true (ByValue, ByPropertyName)
Accept Wildcard Characters?	false

### **-CustomField8<String>**

Specifies a custom field property on an alert object.

Aliases	none
Required?	false
Position?	9
Default Value	none
Accept Pipeline Input?	true (ByValue, ByPropertyName)
Accept Wildcard Characters?	false

### **-CustomField9<String>**

Specifies a custom field property on an alert object.

Aliases	none
Required?	false
Position?	10
Default Value	none
Accept Pipeline Input?	true (ByValue, ByPropertyName)
Accept Wildcard Characters?	false

### **-Owner<String>**

Specifies a value for the Owner property for the alert.

Aliases	none
Required?	false
Position?	13

Default Value	none
Accept Pipeline Input?	true (ByValue, ByPropertyName)
Accept Wildcard Characters?	false

### **-PassThru**

Returns an object representing the alert. By default, this cmdlet does not generate any output.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-ResolutionState<Byte>**

Specifies a resolution state Id. For example, the resolution state Id for "Closed" is 255.

Aliases	none
Required?	false
Position?	12
Default Value	none
Accept Pipeline Input?	true (ByValue, ByPropertyName)
Accept Wildcard Characters?	false

### **-TicketId<String>**

Specifies a value for the TicketId property for the alert.

Aliases	none
Required?	false
Position?	14
Default Value	none

Accept Pipeline Input?	true (ByValue, 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 -----

Description

-----

This command gets all alerts with a resolution state of 15 and then uses the pipeline operator (|) to pass the alert objects to the Set-SCOMAlert cmdlet which closes the alert by setting the resolution state to 255.

```
PS C:\>Get-SCOMAlert -ResolutionState 15 | Set-SCOMAlert -ResolutionState 255
```

### ----- EXAMPLE 2 -----

Description

-----

This command gets all alerts named "Failed Accessing Windows Event Log" and then uses the pipeline operator to pass the alert objects to the Set-SCOMAlert cmdlet which changes the owner and sets the value for CustomField1.

```
PS C:\>Get-SCOMAlert -Name "Failed Accessing Windows Event Log" | Set-SCOMAlert -Owner "CONTOSO\Isabel" -CustomField1 "Root Cause - Permissions"
```

## Related topics

[Get-SCOMAlert](#)

[Get-SCOMConnector](#)

# Set-SCOMAlertResolutionSetting

---

## Set-SCOMAlertResolutionSetting

Sets the alert automatic resolution settings for the management group.

### Syntax

Parameter Set: Empty

```
Set-SCOMAlertResolutionSetting [-AlertAutoResolveDays <Int32> ] [-ComputerName <String[]> ]  
[-Credential <PSCredential> ] [-HealthyAlertAutoResolveDays <Int32> ] [-PassThru] [-  
SCSession <Connection[]> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The Set-SCOMAlertResolutionSetting cmdlet sets the alert automatic resolution settings for the management group.

### Parameters

#### **-AlertAutoResolveDays<Int32>**

The number of days until active alerts in the New monitoring state will automatically be resolved.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByPropertyName)
Accept Wildcard Characters?	false

#### **-ComputerName<String[]>**

Specifies a computer to establish a connection with. The computer must be running the System Center Data Access service. The default is the computer for the current management group connection.

Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-Credential<PSCredential>**

Specifies a user account under which the management group connection will run. It must have access to the server that is specified in the ComputerName parameter if specified. The default is the current user.

Enter a PSCredential object, such as one that is returned by the Get-Credential cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-HealthyAlertAutoResolveDays<Int32>**

Active alerts will automatically be resolved after their alert source is healthy for this many days.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByPropertyName)
Accept Wildcard Characters?	false

### **-PassThru**

If specified, the setting will be returned to the pipeline.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByPropertyName)
Accept Wildcard Characters?	false

### **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection.

Enter a Management group connection object, such as one returned by the Get-SCManagementGroupConnection cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-Confirm**

Prompts you for confirmation before executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#)

## Examples

### ----- EXAMPLE 1 -----

This command sets the alert resolution setting to automatically resolve active alerts in 10 days, and to automatically resolve active alerts after their alert source has been healthy for 5 days.

```
PS C:\>Set-SCOMAlertResolutionSetting -AlertAutoResolveDays 10 -HealthyAlertAutoResolveDays 5
```



# Set-SCOMDatabaseGroomingSetting

---

## Set-SCOMDatabaseGroomingSetting

Sets the database grooming settings for the management group.

### Syntax

Parameter Set: Empty

```
Set-SCOMDatabaseGroomingSetting [-AlertDaysToKeep <Byte> ] [-AvailabilityHistoryDaysToKeep <Byte> ] [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-EventDaysToKeep <Byte> ] [-JobStatusDaysToKeep <Byte> ] [-MaintenanceModeHistoryDaysToKeep <Byte> ] [-MonitoringJobDaysToKeep <Byte> ] [-PassThru] [-PerformanceDataDaysToKeep <Byte> ] [-SCSession <Connection[]> ] [-StateChangeEventDaysToKeep <Byte> ] [-Confirm] [-WhatIf] [<CommonParameters>]
```

### Detailed Description

The Set-SCOMDatabaseGroomingSetting cmdlet sets the database grooming settings for the management group. Database grooming settings are used to automatically remove unnecessary data from the Operations Manager database in order to maintain performance.

### Parameters

#### **-AlertDaysToKeep<Byte>**

The number of days before grooming resolved alerts.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByPropertyName)
Accept Wildcard Characters?	false

#### **-AvailabilityHistoryDaysToKeep<Byte>**

The number of days before grooming availability history.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByPropertyName)
Accept Wildcard Characters?	false

### **-ComputerName<String[]>**

Specifies a computer to establish a connection with. The computer must be running the System Center Data Access service. The default is the computer for the current management group connection.

Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-Credential<PSCredential>**

Specifies a user account under which the management group connection will run. It must have access to the server that is specified in the ComputerName parameter if specified. The default is the current user.

Enter a PSCredential object, such as one that is returned by the Get-Credential cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-EventDaysToKeep<Byte>**

The number of days before grooming event data.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByPropertyName)
Accept Wildcard Characters?	false

## **-JobStatusDaysToKeep<Byte>**

The number of days before grooming task history.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByPropertyName)
Accept Wildcard Characters?	false

## **-MaintenanceModeHistoryDaysToKeep<Byte>**

The number of days before grooming maintenance mode history.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByPropertyName)
Accept Wildcard Characters?	false

## **-MonitoringJobDaysToKeep<Byte>**

The number of days before grooming monitoring job data.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByPropertyName)
Accept Wildcard Characters?	false

## **-PassThru**

If specified, the setting will be returned to the pipeline.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByPropertyName)
Accept Wildcard Characters?	false

## **-PerformanceDataDaysToKeep<Byte>**

The number of days before grooming performance data.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByPropertyName)
Accept Wildcard Characters?	false

## **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection.

Enter a Management group connection object, such as one returned by the Get-SCManagementGroupConnection cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-StateChangeEventDaysToKeep<Byte>**

The number of days before grooming state change 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 -----

This command sets the number of days before grooming resolved alerts to 21.

```
PS C:\>Set-SCOMDatabaseGroomingSetting -AlertDaysToKeep 21
```

### ----- EXAMPLE 2 -----

This command sets the number of days before grooming availability history to 10.

```
PS C:\>Set-SCOMDatabaseGroomingSetting -AvailabilityHistoryDaysToKeep 10
```

# Set-SCOMDataWarehouseSetting

---

## Set-SCOMDataWarehouseSetting

Sets the data warehouse settings for the management group.

### Syntax

Parameter Set: Empty

```
Set-SCOMDataWarehouseSetting [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-DatabaseName <String> ] [-PassThru] [-SCSession <Connection[]> ] [-ServerName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The Set-SCOMDataWarehouseSetting cmdlet sets the data warehouse settings for the management group.

### Parameters

#### **-ComputerName<String[]>**

Specifies a computer to establish a connection with. The computer must be running the System Center Data Access service. The default is the computer for the current management group connection.

Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

#### **-Credential<PSCredential>**

Specifies a user account under which the management group connection will run. It must have access to the server that is specified in the ComputerName parameter if specified. The default is the current user.

Enter a PSCredential object, such as one that is returned by the Get-Credential cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-DatabaseName<String>**

The data warehouse database name.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByPropertyName)
Accept Wildcard Characters?	false

### **-PassThru**

If specified, the setting will be returned to the pipeline.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByPropertyName)
Accept Wildcard Characters?	false

### **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection.



Enter a Management group connection object, such as one returned by the Get-SCManagementGroupConnection cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-ServerName<String>**

The data warehouse server 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 -----

This command sets the data warehouse database to "SCOMDW" and the data warehouse server name to "SCOMServer\INSTANCE1".

```
PS C:\>Set-SCOMDataWarehoueSetting -DatabaseName "SCOMDW" -ServerName "SCOMServer\INSTANCE1"
```

# Set-SCOMErrorReportingSetting

---

## Set-SCOMErrorReportingSetting

Sets the error reporting setting for the management group.

### Syntax

Parameter Set: OptOut

```
Set-SCOMErrorReportingSetting [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-PassThru] [-SCSession <Connection[]> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: NeverSend

```
Set-SCOMErrorReportingSetting -DoNotSend[-ComputerName <String[]> ] [-Credential <PSCredential> ] [-PassThru] [-SCSession <Connection[]> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: Prompt

```
Set-SCOMErrorReportingSetting -PromptBeforeSending[-ComputerName <String[]> ] [-Credential <PSCredential> ] [-PassThru] [-SCSession <Connection[]> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: SendAutomatically

```
Set-SCOMErrorReportingSetting -AutomaticallySend[-ComputerName <String[]> ] [-Credential <PSCredential> ] [-PassThru] [-SCSession <Connection[]> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The Set-SCOMErrorReportingSetting cmdlet sets the error reporting setting for the management group. Error reports are not sent when the DoNotSend parameter is used. Error reports are always sent when the AutomaticallySend parameter is used. Error reports are sent after they have been approved when the PromptBeforeSending parameter is used.

### Parameters

#### -AutomaticallySend

Indicates error reports will automatically be sent without review.

Aliases	none
Required?	true
Position?	named
Default Value	none

Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-ComputerName<String[]>**

Specifies a computer to establish a connection with. The computer must be running the System Center Data Access service. The default is the computer for the current management group connection.

Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-Credential<PSCredential>**

Specifies a user account under which the management group connection will run. It must have access to the server that is specified in the ComputerName parameter if specified. The default is the current user.

Enter a PSCredential object, such as one that is returned by the Get-Credential cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-DoNotSend**

Indicates error reports should not be sent.

Aliases	none
---------	------

Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-PassThru**

If specified, the setting will be returned to the pipeline.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-PromptBeforeSending**

Indicates that error reports need to be approved before sending.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection.

Enter a Management group connection object, such as one returned by the Get-SCManagementGroupConnection cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-Confirm**

Prompts you for confirmation before executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#)

## Examples

### ----- EXAMPLE 1 -----

This command sets the error reporting setting for the management group to always send error reports.

```
PS C:\>Set-SCOMErrorReportingSetting -AutomaticallySend
```

### ----- EXAMPLE 2 -----

This command sets the error reporting setting for the management group to never send error reports.

```
PS C:\>Set-SCOMErrorReportingSetting -DoNotSend
```

### ----- EXAMPLE 3 -----

This command sets the error reporting setting for the management group to only send error reports after they have been approved.

```
PS C:\>Set-SCOMErrorReportingSetting -PromptBeforeSending
```

# Set-SCOMHeartbeatSetting

---

## Set-SCOMHeartbeatSetting

Sets the agent and server heartbeat settings for the management group.

### Syntax

Parameter Set: Empty

```
Set-SCOMHeartbeatSetting [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-HeartbeatInterval <TimeSpan> ] [-MissingHeartbeatThreshold <Int32> ] [-PassThru] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

### Detailed Description

The SCOMHeartbeatSetting cmdlet sets the agent and server heartbeat settings for the management group.

### Parameters

#### **-ComputerName<String[]>**

Specifies a computer to establish a connection with. The computer must be running the System Center Data Access service. The default is the computer for the current management group connection.

Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

#### **-Credential<PSCredential>**

Specifies a user account under which the management group connection will run. It must have access to the server that is specified in the ComputerName parameter if specified. The default is the current user.



Enter a PSCredential object, such as one that is returned by the Get-Credential cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-HeartbeatInterval<TimeSpan>**

The agent heartbeat interval.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByPropertyName)
Accept Wildcard Characters?	false

### **-MissingHeartbeatThreshold<Int32>**

The number of missing heartbeats a management server will ignore before raising an alert.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByPropertyName)
Accept Wildcard Characters?	false

### **-PassThru**

If specified, the setting will be returned to the pipeline.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByPropertyName)
Accept Wildcard Characters?	false

## **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection.

Enter a Management group connection object, such as one returned by the Get-SCManagementGroupConnection cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#)

## **Examples**

### **----- EXAMPLE 1 -----**

This command sets the missing heartbeat threshold on management servers to 5.

```
PS C:\>Set-SCOMHeartbeatSetting -MissingHeartbeatThreshold 5
```

### **----- EXAMPLE 2 -----**

This command sets the agent heartbeat interval to 1 minute.

```
PS C:\>Set-SCOMHeartbeatSetting -HeartbeatInterval "0:01:00"
```



# Set-SCOMLicense

---

## Set-SCOMLicense

Sets the product license level.

### Syntax

Parameter Set: Default

```
Set-SCOMLicense [-ProductId] <String> [-ComputerName <String[]> ] [-Credential  
<PSCredential> ] [-SCSession <Connection[]> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The Set-SCOMLicense cmdlet sets the product license level and removes the evaluation expiration timeout.

**NOTE:** After you enter a license key, restart the System Center Data Access Service on all of your Operations Manager management servers. To view the current license terms, use the [Get-SCOMLicense](#) cmdlet. You can also view the updated license information in the Operations Console Help About dialog box.

### Parameters

#### -ComputerName<String[]>

Specifies a computer to establish a connection with. The computer must be running the System Center Data Access service. The default is the computer for the current management group connection.

Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-Credential<PSCredential>**

Specifies a user account under which the management group connection will run. It must have access to the server that is specified in the ComputerName parameter if specified. The default is the current user.

Enter a PSCredential object, such as one that is returned by the Get-Credential cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-ProductId<String>**

Specifies the product ID.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	true (ByValue, ByPropertyName)
Accept Wildcard Characters?	false

## **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection.

Enter a Management group connection object, such as one returned by the Get-SCManagementGroupConnection cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-Confirm**

Prompts you for confirmation before executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#)

## **Examples**

### **----- EXAMPLE 1 -----**

This command sets the product license level for the product with the ID of "C97A1C5E-6429-4F71-8B2D-3525E237BF62".

```
PS C:\>Set-SCOMLicense -ProductId 'C97A1C5E-6429-4F71-8B2D-3525E237BF62'
```

# Set-SCOMLocation

---

## Set-SCOMLocation

Associates one or more agents with a location.

### Syntax

Parameter Set: FromAgent

```
Set-SCOMLocation [-Agent] <AgentManagedComputer[]> -Location <EnterpriseManagementObject> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-PassThru] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromManagementServer

```
Set-SCOMLocation [-ManagementServer] <ManagementServer[]> -Location <EnterpriseManagementObject> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-PassThru] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromPool

```
Set-SCOMLocation [-Pool] <ManagementServicePool[]> -Location <EnterpriseManagementObject> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-PassThru] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

### Detailed Description

The Set-SCOMLocation cmdlet associates one or more agents with a location so that its state can be displayed in the Web Application Availability Monitoring Summary Map Dashboard. For more information about locations, type: Get-Help New-SCOMLocation -detailed.

### Parameters

#### **-Agent<AgentManagedComputer[]>**

Specifies one or more agent objects. Enter a variable that represents the agents, or type a command that gets the agents.

For information about how to get agents, type Get-Help Get-SCOMAgent.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-ComputerName<String[]>**

Specifies a computer to establish a connection with. The computer must be running the Operations Manager Data Access service.

Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-Credential<PSCredential>**

Specifies a user account under which the management group connection will run. The default is the current user. Type a user name, such as "User01", "Domain01\User01", or "User@Domain.com", or enter a PSCredential object, such as one returned by the Get-Credential cmdlet. When you type a user name, you will be prompted for a password. For more information, type Get-Help Get-Credential.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-Location<EnterpriseManagementObject>**

Specifies a location object.

For information about how to get a location, type Get-Help Get-SCOMLocation.

Aliases	none
Required?	true



Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-ManagementServer<ManagementServer[]>**

Specifies a primary management server object. For information about how to get a management server object, type Get-Help Get-SCOMManagementServer.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-PassThru**

Returns an object representing the location. By default, this cmdlet does not generate any output.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Pool<ManagementServicePool[]>**

Specifies one or more resource pool objects. Enter a variable that represents the resource pools, or type a command that gets the resource pools.

For information about how to get a resource pool object, type "Get-Help Get-SCOMResourcePool".

Aliases	none
---------	------

Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection.

For information about how to get a management group connection, type Get-Help Get-SCOMManagementGroupConnection cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#)

## **Examples**

### **----- EXAMPLE 1 -----**

Description

-----

The first command gets the location object with the display name of Seattle and stores the object in the \$Location variable.

The second command gets the agent object named Server01 and stores the object in the \$Agent variable.

The last command associates the location stored in \$Location to the agent-managed computer stored in \$Agent. If the agent was previously associated with a location, that previous association is deleted.

```
PS C:\> $Location = Get-SCOMLocation -DisplayName "Seattle, WA"
```

```
PS C:\>$Agent = Get-SCOMAgent -Name "Server01.Contoso.com"
```

```
PS C:\>Set-SCOMLocation -Location $Location -Agent $Agent
```

## ----- EXAMPLE 2 -----

### Description

-----

The first command gets the location object with the display name of New York, NY and stores the object in the \$Location variable.

The Second command gets the management server object named MgmtServer01 and stores the object in the \$MgmtServer variable.

The last command associates the location stored in \$Location to the management server stored in \$MgmtServer. If the management server was previously associated with a location, that previous association is deleted.

```
PS C:\> $Location = Get-SCOMLocation -DisplayName "New York, NY"
```

```
PS C:\>$MgmtServer = Get-SCOMManagementServer -Name "MgmtServer01.Contoso.com"
```

```
PS C:\>Set-SCOMLocation -Location $Location -ManagementServer $MgmtServer
```

## ----- EXAMPLE 3 -----

### Description

-----

The first command gets the location object with the display name "Paris, FR" and stores the object in the \$Location variable.

The second command gets the resource pool named Notifications Resource Pool and stores the object in the \$Pool variable.

The last command associates the location stored in \$Location with the resource pool stored in \$Pool. If the resource pool was previously associated with a location, that previous association is deleted.

```
PS C:\> $Location = Get-SCOMLocation -DisplayName "Paris, FR"
```

```
PS C:\>$Pool = Get-SCOMResourcePool -Name "Notifications Resource Pool"PS C:\>Set-SCOMLocation -Location $Location -Pool $Pool
```

## Related topics

[Get-SCOMLocation](#)

[New-SCOMLocation](#)

[Remove-SCOMLocation](#)

# Set-SCOMMaintenanceMode

---

## Set-SCOMMaintenanceMode

Updates active maintenance mode entries.

### Syntax

Parameter Set: FromMaintenanceWindow

```
Set-SCOMMaintenanceMode [-MaintenanceModeEntry] <MaintenanceWindow[]> [-EndTime] <DateTime>
[[-Comment] <String> ] [[-Reason] <MaintenanceModeReason> ] [-PassThru] [
<CommonParameters>]
```

### Detailed Description

The Set-SCOMMaintenanceMode cmdlet updates active maintenance mode entries. This cmdlet only works on active maintenance mode entries.

### Parameters

#### **-Comment<String>**

Allows you to type a comment about the maintenance entry.

Aliases	none
Required?	false
Position?	3
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

#### **-EndTime<DateTime>**

Specifies the time the maintenance will end. The minimum amount of time a resource can be in maintenance mode is 5 minutes. This is a required parameter.

Aliases	none
Required?	true

Position?	2
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-MaintenanceModeEntry<MaintenanceWindow[]>**

Specifies a maintenance mode entry object. Enter a variable that represents the maintenance mode entry, or type a command that gets the maintenance mode entry.

For information about how to get a maintenance mode entry object, type Get-Help Get-SCOMMaintenanceMode.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-PassThru**

Returns an object representing the maintenance mode entry. By default, this cmdlet does not generate any output.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-Reason<MaintenanceModeReason>**

Specifies the reason for placing the resource into maintenance mode. Valid values are: PlannedOther, UnplannedOther, PlannedHardwareMaintenance, UnplannedHardwareMaintenance, PlannedHardwareInstallation, UnplannedHardwareInstallation,

PlannedOperatingSystemReconfiguration, UnplannedOperatingSystemReconfiguration, PlannedApplicationMaintenance, ApplicationInstallation, ApplicationUnresponsive, ApplicationUnstable, SecurityIssue, LossOfNetworkConnectivity.

Aliases	none
Required?	false
Position?	4
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#)

## Examples

### ----- EXAMPLE 1 -----

Description

-----

The first command gets the current time, adds one day (24 hours), and then stores the new time in the \$NewEndTime variable.

The second command gets all class instances in the Contoso.com domain and uses the pipeline operator (|) to pass the class instance objects to the Get-SCOMMaintenance mode cmdlet which gets the maintenance mode entries for each class instance object. Then, the command passes the maintenance mode entry objects to the Set-SCOMMaintenanceMode cmdlet which updates the end time and comment for each maintenance mode entry object.

```
PS C:\>$NewEndTime = (Get-Date).addDays(1)
```

```
PS C:\>Get-SCOMClassInstance -Name *.Contoso.com | Get-SCOMMaintenanceMode | Set-SCOMMaintenanceMode -EndTime $NewEndTime -Comment "Updating end time."
```

### ----- EXAMPLE 2 -----

Description

-----

The first command gets the class instance named "Server01.Contoso.com" and stores it in the \$Instance variable.

The second command gets the maintenance mode entry for the class instance stored in the \$Instance variable, and stores the entry in the \$MMEEntry variable.

The third command gets the current time, adds 30 minutes, and then stores the result in the \$NewEndTime variable.

The fourth command updates the maintenance mode session for the maintenance mode entry stored in the \$MMEntry variable with the end time stored in the \$NewEndTime variable, and the specified comment.

```
PS C:\>$Instance = Get-SCOMClassInstance -Name Server01.Contoso.com
```

```
PS C:\>$MMEntry = Get-SCOMMaintenanceMode -Instance $Instance
```

```
PS C:\>$NewEndTime = (Get-Date).addMinutes(30)
```

```
PS C:\>Set-SCOMMaintenanceMode -MaintenanceModeEntry $MMEntry -EndTime $NewEndTime -Comment  
"Adding 30 minutes to the end time."
```

# Set-SCOMManagementGroupConnection

---

## Set-SCOMManagementGroupConnection

Sets the specified connection as the active connection. The active connection is the connection that is implicitly used when you run a Get- cmdlet without specifying both a –ComputerName parameter and a –Credential parameter, or a –SCSession parameter. Only one connection can be active at any time. By default, the active connection is the last connection that was created by using the New-SCOMManagementGroupConnection cmdlet.

### Syntax

Parameter Set: FromConnection

```
Set-SCOMManagementGroupConnection [-Connection] <Connection> [ <CommonParameters>]
```

### Detailed Description

The Set-SCOMManagementGroupConnection cmdlet sets the specified connection as the active connection. The active connection is the connection that is implicitly used when you run a Get- cmdlet without specifying both a –ComputerName parameter and a –Credential parameter, or a –SCSession parameter. Only one connection can be active at any time. By default, the active connection is the last connection that was created by using the New-SCOMManagementGroupConnection cmdlet.

### Parameters

#### **-Connection<Connection>**

Specifies the connection to activate. You can specify only one connection.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

#### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#)



## Inputs

The input type is the type of the objects that you can pipe to the cmdlet.

- **Microsoft.SystemCenter.Core.Connection.Connection** You can pipe a management group connection to the Connection parameter of the Set-SCOMManagementGroupConnection cmdlet.

## Outputs

The output type is the type of the objects that the cmdlet emits.

- **None.** This cmdlet does not generate any output.

## Examples

### ----- EXAMPLE 1 -----

The commands in this example activate a management group connection. The first command displays the IsActive state of the connection (False), and then the second command activates the connection. The final command verifies the activation by displaying the IsActive state of the connection, which has now changed to True.

```
PS C:\>Get-SCOMManagementGroupConnection
```

```
PS C:\>Get-SCOMManagementGroupConnection | Set-SCOMManagementGroupConnection
```

```
PS C:\>Get-SCOMManagementGroupConnection
```

## Related topics

[Get-SCOMManagementGroupConnection](#)

[New-SCOMManagementGroupConnection](#)

[Remove-SCOMManagementGroupConnection](#)

# Set-SCOMPparentManagementServer

---

## Set-SCOMPparentManagementServer

Changes the primary and failover management servers for an agent or gateway management server.

### Syntax

Parameter Set: FromAgentPrimaryServer

```
Set-SCOMPparentManagementServer [-Agent] <AgentManagedComputer[]> [-PrimaryServer] <ManagementServer> [[-PassThru]] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: FromAgentFailoverServer

```
Set-SCOMPparentManagementServer [-Agent] <AgentManagedComputer[]> [-FailoverServer] <ManagementServer[]> [[-PassThru]] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: FromGatewayFailoverManagementServer

```
Set-SCOMPparentManagementServer [-GatewayServer] <ManagementServer[]> [-FailoverServer] <ManagementServer[]> [[-PassThru]] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: FromGatewayManagementServer

```
Set-SCOMPparentManagementServer [-GatewayServer] <ManagementServer[]> [-PrimaryServer] <ManagementServer> [[-PassThru]] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The Set-SCOMPparentManagementServer cmdlet changes the primary and failover management servers for an agent or gateway management server.

This cmdlet requires an agent object or a gateway management server object, and a management server object. For information about how to get an agent object, type Get-Help Get-SCAgent. For information about how to get a management server object, type Get-Help Get-SCManagementServer. For information about how to get a gateway management server object, type Get-Help Get-SCGatewayManagementServer.

### Parameters

#### **-Agent<AgentManagedComputer[]>**

Specifies one or more agent objects. Enter a variable that represents the agents, or type a command that gets the agents.

Aliases	none
Required?	true
Position?	1
Default Value	none

Accept Pipeline Input?	true (ByValue, ByPropertyName)
Accept Wildcard Characters?	false

### **-FailoverServer<ManagementServer[]>**

Specifies one or more management server objects for the agent to use as failover servers. Enter a variable that represents the failover servers, or type a command that gets the failover servers.

For information about how to get a management server object, type Get-Help Get-SCManagementServer.

Aliases	none
Required?	true
Position?	2
Default Value	none
Accept Pipeline Input?	true (ByValue, ByPropertyName)
Accept Wildcard Characters?	false

### **-GatewayServer<ManagementServer[]>**

Specifies one or more gateway management server objects. Enter a variable that represents the gateway management servers, or type a command that gets the gateway management servers.

For information about how to get a management server object, type Get-Help Get-SCGatewayManagementServer.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	true (ByValue, ByPropertyName)
Accept Wildcard Characters?	false

### **-PassThru**

Returns an object representing the updated settings. By default, this cmdlet does not generate any output.

Aliases	none
Required?	false
Position?	3
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-PrimaryServer<ManagementServer>**

Specifies the management server object for the agent to use as its primary server. Enter a variable that represents the primary management server, or type a command that gets the primary management server.

Aliases	none
Required?	true
Position?	2
Default Value	none
Accept Pipeline Input?	true (ByValue, ByPropertyName)
Accept Wildcard Characters?	false

### **-Confirm**

Prompts you for confirmation before executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

Required?	false
-----------	-------

Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#)

## Examples

### ----- EXAMPLE 1 -----

The first two commands get a management server to set as the agent's primary management server, and a management server to set as the agent's failover management server. The commands store the objects in the \$PrimaryMgmtServer, and \$FailoverMgmtServer variables, respectively.

The third command gets the agent named "Server01.Contoso.com" and sets the primary and failover management servers specified in the \$PrimaryMgmtServer and \$FailoverMgmtServer variables. It then uses the PassThru parameter to generate an object. Without the PassThru parameter, Set-SCOMPparentManagementServer does not generate any output.

```
PS C:\>$PrimaryMgmtServer = Get-SCManagementServer -Name "MgmtServer01.Contoso.com"PS
C:\>$FailoverMgmtServer = Get-SCManagementServer -Name "MgmtServer02.Contoso.com"PS
C:\>"Server01.Contoso.com" | Get-SCAgent | Set-SCOMPparentManagementServer -PrimaryServer
$PrimaryMgmtServer -FailoverServer $FailoverMgmtServer -Passthru
```

### ----- EXAMPLE 2 -----

The first two commands get a management server to set as the gateway server's primary management server, and a management server to set as the gateway server's failover management server. The commands store the objects in the \$PrimaryMgmtServer, and \$FailoverMgmtServer variables, respectively.

The third command gets the gateway management server named "GatewayMgmtServer01.Contoso.com" and sets the primary and failover management servers specified in the \$PrimaryMgmtServer and \$FailoverMgmtServer variables for the gateway management server.

```
PS C:\>$PrimaryMgmtServer = Get-SCManagementServer -name "MgmtServer01.Contoso.com"PS
C:\>$FailoverMgmtServer = Get-SCManagementServer -Name "MgmtServer02.Contoso.com"PS
C:\>"GatewayMgmtServer01.Contoso.com" | Get-SCGatewayManagementServer | Set-
SCOMPparentManagementServer -PrimaryServer $PrimaryMgmtServer -FailoverServer
$FailoverMgmtServer
```

### ----- EXAMPLE 3 -----

This command sets the primary and failover management servers for the specified agent.

The commands in parentheses, which are executed first, get the agent named Server01, and the management server named MgmtServer01. The cmdlet then passes the results of the commands in parentheses to Set-SCOMPARENTMANAGEMENTSERVER. Set-SCOMPARENTMANAGEMENTSERVER then sets the primary server for the agent.

```
PS C:\>Set-SCOMPARENTMANAGEMENTSERVER -Agent (Get-SCAGENT -Name "Server01.Contoso.com") -  
PrimaryServer (Get-SCMANAGEMENTSERVER -Name "MgmtServer01.Contoso.com")
```

# Set-SCOMReportingSetting

---

## Set-SCOMReportingSetting

Sets the reporting server for the management group.

### Syntax

Parameter Set: Empty

```
Set-SCOMReportingSetting [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-PassThru] [-ReportingServerUrl <Uri> ] [-SCSession <Connection[]> ] [-Confirm] [-WhatIf] [<CommonParameters>]
```

### Detailed Description

The Set-SCOMReportingSetting cmdlet sets the reporting server for the management group.

### Parameters

#### **-ComputerName<String[]>**

Specifies a computer to establish a connection with. The computer must be running the System Center Data Access service. The default is the computer for the current management group connection.

Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

#### **-Credential<PSCredential>**

Specifies a user account under which the management group connection will run. It must have access to the server that is specified in the ComputerName parameter if specified. The default is the current user.

Enter a PSCredential object, such as one that is returned by the Get-Credential cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-PassThru**

If specified, the setting will be returned to the pipeline.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByPropertyName)
Accept Wildcard Characters?	false

### **-ReportingServerUri<Uri>**

Specifies the reporting server URL.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByPropertyName)
Accept Wildcard Characters?	false

### **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection.



Enter a Management group connection object, such as one returned by the Get-SCManagementGroupConnection cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-Confirm**

Prompts you for confirmation before executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#)

## Examples

### ----- EXAMPLE 1 -----

This command sets the reporting server URL for the management group to `http://Reporting.Contoso.com`.

```
PS C:\>Set-SCOMReportingSetting -ReportingServerUrl "http://Reporting.Contoso.com"
```

# Set-SCOMResourcePool

---

## Set-SCOMResourcePool

Changes the properties of a resource pool.

### Syntax

Parameter Set: Empty

```
Set-SCOMResourcePool [-ResourcePool] <ManagementServicePool[]> [-ComputerName <String[]> ]  
[-Credential <PSCredential> ] [-PassThru] [-SCSession <Connection[]> ] [-Confirm] [-WhatIf]  
[ <CommonParameters>]
```

Parameter Set: FromAutoPopulate

```
Set-SCOMResourcePool [-ResourcePool] <ManagementServicePool[]> [-EnableAutomaticMembership]  
<Boolean> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-PassThru] [-SCSession  
<Connection[]> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: FromMember

```
Set-SCOMResourcePool [-ResourcePool] <ManagementServicePool[]> [-Member]  
<ComputerHealthService[]> [-Action <UpdateAction> ] [-ComputerName <String[]> ] [-Credential  
<PSCredential> ] [-PassThru] [-SCSession <Connection[]> ] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: FromObserver

```
Set-SCOMResourcePool [-ResourcePool] <ManagementServicePool[]> [-Observer]  
<ComputerHealthService[]> [-Action <UpdateAction> ] [-ComputerName <String[]> ] [-Credential  
<PSCredential> ] [-PassThru] [-SCSession <Connection[]> ] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

### Detailed Description

The Set-SCOMResourcePool cmdlet changes the properties of a specified resource pool.

### Parameters

#### **-Action<UpdateAction>**

Specifies the action to take when updating the members of a resource pool. Valid values include "Add" or "Remove".

Aliases	none
Required?	true
Position?	3
Default Value	none

Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-ComputerName<String[]>**

Specifies the name of a computer to establish a connection with. The computer must be running the System Center Data Access Service. The default value is the computer for the current management group connection. Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-Credential<PSCredential>**

Specifies a user account under which the management group connection will run. The default is the current user. Type a user name, such as "User01", "Domain01\User01", or "User@Domain.com", or enter a PSCredential object, such as one returned by the Get-Credential cmdlet. When you type a user name, you will be prompted for a password. For more information, type Get-Help Get-Credential.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-EnableAutomaticMembership<Boolean>**

Enables automatic membership for the resource pool. When set to \$True, the resource pool contains all management servers, and membership in the pool is automatically managed. When set to \$False, the resource pool can contain management servers, gateway management servers, or both, and membership in the pool is manually managed.

If the value is changed from \$True to \$False, the membership of the pool remains the same until manually updated.

If the value is changed from \$False to \$True, all members of the pool are removed, and then the pool is automatically re-populated with all management servers. The resource pool will be automatically updated as management servers are added and deleted from Operations Manager. This process can take some time to complete.

Aliases	none
Required?	true
Position?	2
Default Value	none
Accept Pipeline Input?	true (ByPropertyName)
Accept Wildcard Characters?	false

### **-Member<ComputerHealthService[]>**

Specifies an object to include in the resource pool.

Valid objects that can be members of a resource pool include management servers and gateway servers. For information about how to get a management server object, type "Get-Help Get-SCOMManagementServer". For information about how to get a gateway server object, type "Get-Help Get-SCOMGatewayManagementServer".

Aliases	none
Required?	true
Position?	2
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-Observer<ComputerHealthService[]>**

Specifies a management server or a gateway management server that is not a member of the resource pool.

To make a resource pool highly available, you must add a minimum of three members to the pool, or two members and one observer.

Aliases	none
---------	------

Required?	true
Position?	2
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-PassThru**

Returns an object representing the resource pool. By default, this cmdlet does not generate any output.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-ResourcePool<ManagementServicePool[]>**

Specifies one or more resource pool objects. Enter a variable that represents the resource pools, or type a command that gets the resource pools.

For information about how to get a resource pool object, type "Get-Help Get-SCOMResourcePool".

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection. Specify a Management group connection object, such as one returned by the Get-SCOMManagementGroupConnection cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-Confirm**

Prompts you for confirmation before executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#)

## Examples

### ----- EXAMPLE 1 -----

#### Description

-----

The first command gets all management servers and stores them in the \$Members variable.

The second command gets the resource pool with a display name of "Pool01" and then adds the objects stored in the \$Members variable to that resource pool.

```
PS C:\> $Members = Get-SCOMManagementServer
```

```
PS C:\>Get-SCOMResourcePool -DisplayName "Pool01" | Set-SCOMResourcePool -Member $Members -  
Action "Add"
```

### ----- EXAMPLE 2 -----

#### Description

-----

The first command gets the management server with the display name of "Member01" and stores it in the \$Member variable.

The second command gets the resource pool with the display name of "Pool01" and then removes the object stored in the \$Member variable from that resource pool.

```
PS C:\> $Member = Get-SCOMManagementServer -Name "Member01"
```

```
PS C:\>Get-SCOMResourcePool -DisplayName Pool01 | Set-SCOMResourcePool -Member $Member -  
Action "Remove"
```

### ----- EXAMPLE 3 -----

#### Description

-----

The first command gets the management server named Observer01.

The second command gets the resource pool with the display name of "Pool01" and then adds the Observer object stored in the \$Observer variable to that resource pool.

```
PS C:\> $Observer = Get-SCOMManagementServer -Name "Observer01"
```

```
PS C:\>Get-SCOMResourcePool -DisplayName Pool01 | Set-SCOMResourcePool -Observer $Observer  
-Action "Add" -Passthru
```

### ----- EXAMPLE 4 -----

#### Description

-----

The first command gets the management server named Observer01.



The second command gets the resource pool with the display name of "Pool01" and then removes the Observer object stored in the \$Observer variable from that resource pool.

```
PS C:\> $Observer = Get-SCOMManagementServer -Name "Observer01"
```

```
PS C:\>Get-SCOMResourcePool -DisplayName Pool01 | Set-SCOMResourcePool -Observer $Observer -  
Action "Remove" -Passthru
```

## ----- EXAMPLE 5 -----

Description

-----

The first command gets the resource pool with a display name of Pool01.

The second command changes the value of the DisplayName property for the resource pool stored in the \$Pool variable to "New Pool 02".

The last command uses the ApplyChanges() method to commit the change to the DisplayName property.

```
PS C:\> $Pool = Get-SCOMResourcePool -DisplayName "Pool01"
```

```
PS C:\>$Pool.DisplayName = "New Pool 02"
```

```
PS C:\>$Pool.ApplyChanges()
```

## ----- EXAMPLE 6 -----

Description

-----

The first command gets the resource pool with a display name of New Pool 02.

The second command changes the value of the Description property for the resource pool stored in the \$Pool variable.

The last command uses the ApplyChanges() method to commit the change to the Description property.

```
PS C:\> $Pool = Get-SCOMResourcePool -DisplayName "New Pool 02"
```

```
PS C:\>$Pool.Description = "Description of New Pool 02"
```

```
PS C:\>$Pool.ApplyChanges()
```

## Related topics

[Get-SCResourcePool](#)

[New-SCResourcePool](#)

[Remove-SCResourcePool](#)

# Set-SCOMRMSEmulator

---

## Set-SCOMRMSEmulator

Moves the RMS Emulator role to the specified management server.

### Syntax

Parameter Set: Empty

```
Set-SCOMRMSEmulator [-Server] <ManagementServer> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-PassThru] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

### Detailed Description

The Set-SCOMRMSEmulator cmdlet creates the RMS Emulator role on the specified management server or moves it to this server if the role already exists in the management group.

### Parameters

#### **-ComputerName<String[]>**

Specifies the name of a computer to establish a connection with. The computer must be running the System Center Data Access Service. The default value is the computer for the current management group connection. Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

Aliases	none
Required?	false
Position?	named
Default Value	localhost
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

#### **-Credential<PSCredential>**

Specifies a user account under which the management group connection will run. The default is the current user. Type a user name, such as "User01", "Domain01\User01", or "User@Domain.com", or enter a PSCredential object, such as one returned by the Get-Credential cmdlet. When you type a user name, you will be prompted for a password. For more information, type Get-Help Get-Credential.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-PassThru**

Returns an object representing the RMS Emulator role. By default, this cmdlet does not generate any output.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection. Specify a Management group connection object, such as one returned by the Get-SCOMManagementGroupConnection cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-Server<ManagementServer>**

Specifies the management server for the RMS Emulator role.

For information about how to get a management server, type `Get-Help Get-SCOMManagementServer`.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	true (ByValue, 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 -----**

Description

-----

This command creates the RMS Emulator role on the management server `Server01.Contoso.com`.

```
PS C:\>Get-SCOMManagementServer -Name "Server01.Contoso.com" | Set-SCOMRMSEmulator
```

## **Related topics**

[Get-SCOMManagementServer](#)

[Get-SCOMRMSEmulator](#)

[Remove-SCOMRMSEmulator](#)

# Set-SCOMRunAsDistribution

---

## Set-SCOMRunAsDistribution

Sets the distribution policy of an Operations Manager Run As account.

### Syntax

Parameter Set: LessSecure

```
Set-SCOMRunAsDistribution [-RunAsAccount] <SecureData> -LessSecure[-ComputerName <String[]> ] [-Credential <PSCredential> ] [-PassThru] [-SCSession <Connection[]> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: MoreSecure

```
Set-SCOMRunAsDistribution [-RunAsAccount] <SecureData> -MoreSecure[-ComputerName <String[]> ] [-Credential <PSCredential> ] [-PassThru] [-SCSession <Connection[]> ] [-SecureDistribution <Object[]> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: Security

```
Set-SCOMRunAsDistribution [-RunAsAccount] <SecureData> -Security <String> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-PassThru] [-SCSession <Connection[]> ] [-SecureDistribution <Object[]> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

## Detailed Description

The Set-SCOMRunAsDistribution cmdlet sets the distribution policy of an Operations Manager Run As account. Distribution policies determine which computers receive a Run As account credential. New accounts have the "more secure" distribution by default, with no approved systems.

## Parameters

### -ComputerName<String[]>

Specifies a computer to establish a connection with. The computer must be running the System Center Data Access service. The default is the computer for the current management group connection.

Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false

Accept Wildcard Characters?	false
-----------------------------	-------

### **-Credential<PSCredential>**

Specifies a user account under which the management group connection will run. It must have access to the server that is specified in the ComputerName parameter if specified. The default is the current user.

Enter a PSCredential object, such as one that is returned by the Get-Credential cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-LessSecure**

If set, the credential will be distributed automatically to all managed computers.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-MoreSecure**

If set, the credential will only be distributed to systems specified in the SecureDistribution parameter

Aliases	none
Required?	true
Position?	named
Default Value	none

Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-PassThru**

If set, returns the Run As account distribution policy.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-RunAsAccount<SecureData>**

The Run As account for which distribution will be set. Enter a Microsoft.EnterpriseManagement.Security.SecureData object, such as one returned from Get-SCOMRunAsAccount.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection.

Enter a Management group connection object, such as one returned by the Get-SCManagementGroupConnection cmdlet.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-SecureDistribution<Object[]>**

The list of systems which to which the account will be authorized for distribution.

This list may contain only the following types of items:

-- Agents - Objects returned from Get-SCOMAgent. The account will be authorized for distribution to the agent.

-- Management Servers - Objects returned from Get-SCOMManagementServer. The account will be authorized for distribution to the server.

-- Pools - Objects returned from Get-SCOMResourcePool. The account will be authorized for distribution to all members of the pool.

-- Healthservice Instances - Objects returned from Get-SCOMClassInstance, which have a managed type of HealthService. The account will be authorized for distribution to this healthservice.

This parameter is automatically populated when piped input from Get-SCOMRunAsDistribution.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByPropertyName)
Accept Wildcard Characters?	false

### **-Security<String>**

Either "MoreSecure" or "LessSecure". This parameter will be populated automatically if piping input from Get-SCOMRunAsDistribution.

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 -----**

This command sets the contoso\lowpriv account for "less secure" distribution.

```
PS C:\>Get-SCOMRunAsAccount "contoso\lowpriv" | Set-SCOMRunAsDistribution -LessSecure
```

## ----- EXAMPLE 2 -----

This command sets the contoso\highpriv account for "more secure" distribution, with no approved systems.

```
PS C:\>Get-SCOMRunAsAccount "contoso\highpriv " | Set-SCOMRunAsDistribution -MoreSecure
```

## ----- EXAMPLE 3 -----

This example sets the contoso\monitoring account for "more secure" distribution to a collection of pools, agents, and servers.

```
PS C:\>$distribution = ( Get-SCOMAgent -Name "*.contoso.com" ) + ( Get-SCOMManagementServer ) + ( Get-SCOMResourcePool -DisplayName "Contoso Monitoring Pool" )
```

```
PS C:\>Get-SCOMRunAsAccount "contoso\lowpriv" | Set-SCOMRunAsDistribution -MoreSecure -SecureDistribution $distribution
```

## ----- EXAMPLE 4 -----

This command creates a new Windows Run As account and approves it for distribution to all agents.

```
PS C:\>Add-SCOMRunAsAccount -Windows -Name "NewAccount" -Credential (Get-Credential) | Set-SCOMRunAsDistribution -MoreSecure -SecureDistribution (Get-SCOMAgent)
```

## ----- EXAMPLE 5 -----

This example copies the secure distribution policy from the contoso\administrator account to the contoso\monitoring account.

```
PS C:\>$monitoringAcct = Get-SCOMRunAsAccount "contoso\monitoring"
```

```
PS C:\>Get-SCOMRunAsAccount "contoso\administrator" | Get-SCOMRunAsDistribution | Set-SCOMRunAsDistribution -RunAsAccount $monitoringAccount
```

# Set-SCOMRunAsProfile

---

## Set-SCOMRunAsProfile

Adds Run As accounts to or removes them from a Run As profile.

### Syntax

Parameter Set: Empty

```
Set-SCOMRunAsProfile [-Account] <SecureData[]> [-Profile] <ManagementPackSecureReference> [-Action] <RunAsProfileAccountsAction> [-PassThru] [ <CommonParameters>]
```

Parameter Set: FromGroup

```
Set-SCOMRunAsProfile [-Group] <MonitoringObject[]> [-Account] <SecureData[]> [-Profile] <ManagementPackSecureReference> [-Action] <RunAsProfileAccountsAction> [-PassThru] [ <CommonParameters>]
```

Parameter Set: FromInstance

```
Set-SCOMRunAsProfile [-Instance] <MonitoringObject[]> [-Account] <SecureData[]> [-Profile] <ManagementPackSecureReference> [-Action] <RunAsProfileAccountsAction> [-PassThru] [ <CommonParameters>]
```

Parameter Set: FromManagementPackClass

```
Set-SCOMRunAsProfile [-Class] <ManagementPackClass[]> [-Account] <SecureData[]> [-Profile] <ManagementPackSecureReference> [-Action] <RunAsProfileAccountsAction> [-PassThru] [ <CommonParameters>]
```

### Detailed Description

The Set-SCOMRunAsProfile cmdlet adds Run As accounts to or removes Run As accounts from a Run As profile.

### Parameters

#### **-Account<SecureData[]>**

Specifies the Run As account to add to the Run As profile. Enter a variable that represents the Run As account, or type a command that gets the account.

For information about how to get a Run As account, type Get-Help Get-SCOMRunAsAccount.

Aliases	none
Required?	true
Position?	2
Default Value	none

Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-Action<RunAsProfileAccountsAction>**

Specifies the action to take when updating the Run As profile. Valid values include "Add" or "Remove".

Aliases	none
Required?	true
Position?	4
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-Class<ManagementPackClass[]>**

Specifies the classes that the Run As account is permitted to manage. Enter a variable that represents the class, or type a command that gets the class. For information about how to get a class object, type Get-Help Get-SCOMClass.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-Group<MonitoringObject[]>**

Specifies the groups that the Run As account is permitted to manage. Enter a variable that represents the group, or type a command that gets the group.

For information about how to get a class object, type Get-Help Get-SCOMGroup.

Aliases	none
Required?	true

Position?	1
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-Instance<MonitoringObject[]>**

Specifies the class instances that the Run As account is permitted to manage. Enter a variable that represents the class instance, or type a command that gets the class instance. This parameter also accepts group objects.

For information about how to get a class object, type Get-Help Get-SCOMClassInstance.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-PassThru**

Returns an object representing the Run as profile. By default, this cmdlet does not generate any output.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Profile<ManagementPackSecureReference>**

Specifies the Run As profile to update. Enter a variable that represents the Run As profile, or type a command that gets the profile.

For information about how to get a Run As profile, type Get-Help Get-SCOMRunAsProfile.

Aliases	none
Required?	true
Position?	3
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#)

## Examples

### ----- EXAMPLE 1 -----

Description

-----

The first command gets the Run As profile with the display name "Privileged Monitoring Account" and stores it in the \$Profile variable.

The second command gets the Run As account named "High Privileged Account" and stores it in the \$Account variable.

The third command adds the account stored in the \$Account variable to the Run As profile stored in the \$Profile variable, and configures the Run As account to manage "All targeted objects" because no class, group or object was provided.

```
PS C:\> $Profile = Get-RunAsProfile -DisplayName "Privileged Monitoring Account"
```

```
PS C:\>$Account = Get-SCOMRunAsAccount -Name "High Privileged Account"
```

```
PS C:\>Set-SCOMRunAsProfile -Action "Add" -Profile $Profile -Account $Account
```

### ----- EXAMPLE 2 -----

Description

-----

The first command gets the Run As profile named "SQL Server Monitoring Account" and stores it in the \$Profile variable.

The second command gets the Run As account named "Contoso\SQLAuth" and stores it in the \$Account variable.

The third command gets the group named "Contoso financial SQL Servers" and stores it in the \$Group variable.

The fourth command adds the account stored in the \$Account variable to the Run As profile stored in the \$Profile variable, and configures the Run As account to manage the group stored in the \$Group variable.

```
PS C:\> $Profile = Get-SCOMRunAsProfile -DisplayName "SQL Server Monitoring Account"
PS C:\>$Account = Get-SCOMrunAsAccount -Name "Contoso\SQLAuth"
PS C:\>$Group = Get-SCOMGroup -DisplayName "Contoso financial SQL Servers"
PS C:\>Set-SCOMRunAsProfile -Action "Add" -Profile $Profile -Account $Account -Group $Group
```

### ----- EXAMPLE 3 -----

#### Description

-----

The first command gets the Run As profile object with the display name SQL Server Monitoring Account and stores the object in the \$Profile variable.

The second command gets the Run As account object named Contoso\SQLAuth and stores the object in the \$Account variable.

The third command gets the group object with the display name Contoso financial SQL Servers and stores the object in the \$Group variable.

The last command adds the account and group stored in \$Account and \$Group to the profile SQL Server Monitoring Account (stored in \$Profile).

```
PS C:\> $Profile = Get-SCOMRunAsProfile -DisplayName "SQL Server Monitoring Account"
PS C:\>$Account = Get-SCOMrunAsAccount -Name "Contoso\SQLAuth"
PS C:\>$Group = Get-SCOMGroup -DisplayName "Contoso financial SQL Servers"
PS C:\>Set-SCOMRunAsProfile -Action "Add" -Profile $Profile -Account $Account -Group $Group
```

## Related topics

[Get-SCOMRunAsAccount](#)

[Get-SCOMClass](#)

[Get-SCOMClassInstance](#)

[Get-SCOMGroup](#)

[Get-SCOMRunAsProfile](#)

[Get-SCOMProfile](#)

# Set-SCOMUserRole

---

## Set-SCOMUserRole

Configures an Operations Manager user role.

### Syntax

Parameter Set: FromAllClassScope

```
Set-SCOMUserRole [-UserRole] <UserRole> -AllClass[-PassThru] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: FromAllGroupScope

```
Set-SCOMUserRole [-UserRole] <UserRole> -AllGroup[-PassThru] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: FromAllTaskScope

```
Set-SCOMUserRole [-UserRole] <UserRole> -AllTask[-PassThru] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

Parameter Set: FromClassScope

```
Set-SCOMUserRole [-UserRole] <UserRole> -ClassScope <ManagementPackClass[]> [-PassThru] [-  
Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: FromGroupscope

```
Set-SCOMUserRole [-UserRole] <UserRole> -GroupScope <MonitoringObjectGroup[]> [-PassThru] [-  
Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: FromTaskScope

```
Set-SCOMUserRole [-UserRole] <UserRole> -TaskScope <ManagementPackTask[]> [-PassThru] [-  
Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: FromUser

```
Set-SCOMUserRole [-UserRole] <UserRole> -User <String[]> [-PassThru] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

### Detailed Description

The Set-SCOMUserRole cmdlet configures an Operations Manager user role. You can update the user list or scope of the user role.

### Parameters

#### -AllClass

Indicates that an Author user role is scoped to all classes.

Aliases	none
Required?	true



Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-AllGroup**

Indicates that a user role is scoped to all groups.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-AllTask**

Indicates that an Author, Operator, or Advanced Operator user rule is scoped to all tasks.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-ClassScope<ManagementPackClass[]>**

Specifies the new collection of classes that an Author user role is scoped to. To remove all scoped classes, specify \$null or an empty array @().

Aliases	none
Required?	true

Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-GroupScope<MonitoringObjectGroup[]>**

Specifies the new collection of groups that a user role is scoped to. To remove all scoped groups, specify \$null or an empty array @().

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-PassThru**

Returns an object representing the user role. By default, this cmdlet does not generate any output.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-TaskScope<ManagementPackTask[]>**

Specifies the new collection of tasks that an Author, Operator, or Advanced Operator user role is scoped to. To remove all scoped tasks, specify \$null or an empty array @().

Aliases	none
Required?	true

Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-User<String[]>**

Specifies one or more user names to include in the user role. This list replaces any existing list of users. To clear all users from a user role, specify \$null or an empty array @().

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-UserRole<UserRole>**

Specifies a user role object.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-Confirm**

Prompts you for confirmation before executing the command.

Required?	false
Position?	named

Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## -WhatIf

Describes what would happen if you executed the command without actually executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#)

## Examples

### ----- EXAMPLE 1 -----

The first command gets the user role object named Contoso Operators and stores the object in the \$role variable.

The second command uses the pipeline operator to pass the user role stored in \$role to Set-SCOMUserRole which grants the user role access to all tasks and returns the updated user role object. The command then passes the updated object to Set-SCOMUserRole which approves the role for all groups.

```
PS C:\> $role = Get-SCOMUserRole -Name 'Constoso Operators'
```

```
PS C:\>$role | Set-SCOMUserRole -AllTask -PassThru | Set-SCOMUserRole -AllGroup
```

### ----- EXAMPLE 2 -----

The first command gets the user role object named Contoso SQL Operators and stores the object in the \$role variable.

The second command gets all task objects with SQL in their name and stores the objects in the \$newTaskList variable

The last command uses the pipeline operator to pass the user role stored in \$role to Set-SCOMUserRole which resets the collection of approved tasks to the list stored in \$newTaskList

```
PS C:\> $role = Get-SCOMUserRole -Name 'Constoso SQL Operators'
```

```
PS C:\>$newTaskList = Get-SCOMTask -Name '*SQL*'
```

```
PS C:\>$role | Set-SCOMUserRole -TaskScope $newTaskList
```

### ----- EXAMPLE 3 -----

The first command gets the user role object named Contoso Read-Only Operators and stores the object in the \$role variable.

The second command uses the pipeline operator to pass the role stored in \$role to Set-SCOMUserRole which adds the user contoso\Cesar to the existing list of users for the role.

```
PS C:\> $role = Get-SCOMUserRole -Name 'Contoso Read-Only Operators'
```

```
PS C:\>$role | Set-SCOMUserRole -User ($role.Users + 'contoso\Cesar')
```

# Set-SCOMWebAddressSetting

---

## Set-SCOMWebAddressSetting

Sets the web console and online product knowledge URLs for the management group.

### Syntax

Parameter Set: Empty

```
Set-SCOMWebAddressSetting [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-OnlineProductKnowledgeUrl <Uri> ] [-PassThru] [-SCSession <Connection[]> ] [-WebConsoleUrl <Uri> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The Set-SCOMWebAddressSetting cmdlet sets the web console and online product knowledge URLs for the management group.

### Parameters

#### **-ComputerName<String[]>**

Specifies a computer to establish a connection with. The computer must be running the System Center Data Access service. The default is the computer for the current management group connection.

Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

#### **-Credential<PSCredential>**

Specifies a user account under which the management group connection will run. It must have access to the server that is specified in the ComputerName parameter if specified. The default is the current user.

Enter a PSCredential object, such as one that is returned by the Get-Credential cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-OnlineProductKnowledgeUri<Uri>**

The URL for online product knowledge.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByPropertyName)
Accept Wildcard Characters?	false

### **-PassThru**

If specified, the setting will be returned to the pipeline.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByPropertyName)
Accept Wildcard Characters?	false

### **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection.

Enter a Management group connection object, such as one returned by the Get-SCManagementGroupConnection cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-WebConsoleUrl<Uri>**

The URL for the web console.

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 -----

This command sets the web console URL to "http://OM01.contoso.com/OperationsManager".

```
PS C:\>Set-SCOMWebAddressSetting -WebConsoleUrl "http://OM01.contoso.com/OperationsManager"
```

### ----- EXAMPLE 2 -----

This command sets the online product knowledge URL to "http://OM01.contoso.com/Knowledge".

```
PS C:\>Set-SCOMWebAddressSetting -OnlineProductKnowledgeUrl  
"http://OM01.contoso.com/Knowledge"
```

# Start-SCOMMaintenanceMode

---

## Start-SCOMMaintenanceMode

Places a specified class instance into maintenance mode, and creates new active maintenance mode entries.

### Syntax

Parameter Set: FromInstance

```
Start-SCOMMaintenanceMode [-Instance] <MonitoringObject[]> [-EndTime] <DateTime> [[-Comment] <String> ] [[-Reason] <MaintenanceModeReason> ] [-PassThru] [ <CommonParameters>]
```

### Detailed Description

The Start-MaintenanceMode cmdlet places a specified class instance into maintenance mode, and creates new active maintenance mode entries.

When in maintenance mode, alerts, notifications, rules, monitors, automatic responses, state changes, and new alerts are suppressed for the class instance.

### Parameters

#### **-Comment<String>**

Allows you to type a comment about the maintenance activity.

Aliases	none
Required?	false
Position?	3
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

#### **-EndTime<DateTime>**

Specifies the time that the maintenance will end. The minimum amount of time a resource can be in maintenance mode is 5 minutes.

Aliases	none
Required?	true
Position?	2
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-Instance<MonitoringObject[]>**

Specifies a class instance object. Enter a variable that represents the class instance, or type a command that gets the class instance. This parameter also accepts group objects.

For information about how to get a class instance object, type `Get-Help Get-SCOMClassInstance`.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-PassThru**

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Reason<MaintenanceModeReason>**

Specifies the reason for placing the resource into maintenance mode. Valid values are: `PlannedOther`, `UnplannedOther`, `PlannedHardwareMaintenance`, `UnplannedHardwareMaintenance`, `PlannedHardwareInstallation`, `UnplannedHardwareInstallation`,

PlannedOperatingSystemReconfiguration, UnplannedOperatingSystemReconfiguration, PlannedApplicationMaintenance, ApplicationInstallation, ApplicationUnresponsive, ApplicationUnstable, SecurityIssue, LossOfNetworkConnectivity.

Aliases	none
Required?	false
Position?	4
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#)

## Examples

### ----- EXAMPLE 1 -----

Description

-----

The first command gets the class instance named Server01.Contoso.com.

The second command gets the current time and adds ten minutes.

The third command starts maintenance on the object stored in the \$Instance variable, sets the end time for the time stored in the \$Time variable, and specifies a reason and a comment about the maintenance mode activity.

```
PS C:\>$Instance = Get-SCOMClassInstance -Name Server01.Contoso.com
```

```
PS C:\>$Time = ((Get-Date).AddMinutes(10))
```

```
PS C:\>Start-SCOMMaintenanceMode -Instance $Instance -EndTime $Time -Reason "SecurityIssue"  
-Comment "Applying software update."
```

## Related topics

[Get-SCOMClassInstance](#)

[Get-SCOMMaintenanceMode](#)

[Set-SCOMMaintenanceMode](#)

# Start-SCOMTask

---

## Start-SCOMTask

Starts a task against a specified object.

### Syntax

Parameter Set: Empty

```
Start-SCOMTask [-Instance] <EnterpriseManagementObject[]> [-Task] <ManagementPackTask> [[-TaskCredentials] <PSCredential> ] [[-Override] <Hashtable> ] [ <CommonParameters>]
```

### Detailed Description

The Start-Task cmdlet starts a task against a specified object. This cmdlet requires a task object and a class instance object. This cmdlet accepts only one task, but can accept multiple class instances and overrides.

### Parameters

#### **-Instance<EnterpriseManagementObject[]>**

Specifies one or more class instance objects for which to start a task. Enter a variable that represents the class instances, or type a command that gets the class instances. This parameter also accepts group objects.

For information about how to get a class instance object, type Get-Help Get-SCOMClassInstance.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

#### **-Override<Hashtable>**

Specifies a hash table that defines the values for one or more overridable parameters of the task being run. To determine the overridable parameters for a task, use the GetOverridableParameters() method for a task object.

A hash table is a compact data structure that stores one or more name/value pairs. To create a hash table that defines the values for overridable parameters, use the following syntax:

```
@{  
Parameter1=10;  
Parameter2="1092834098123407953912837"}
```

To use a hash table in a command or script, save it in a variable. For example:

```
PS C:\> $Overrides = @{  
>> Parameter1=10;  
>> Parameter2="1092834098123407953912837"  
>> }
```

For more information about hash tables, type `Get-Help about_Hash_Tables`.

Aliases	none
Required?	false
Position?	4
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-Task<ManagementPackTask>**

Specifies a task object to start. Enter a variable that represents the task, or type a command that gets the task. This parameter accepts only one task.

For information about how to get a task object, type `Get-Help Get-SCOMTask`.

Aliases	none
Required?	true
Position?	2
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-TaskCredentials<PSCredential>**

Specifies the credentials under which the task will run. If this parameter is omitted, the current user's credentials are used.

Aliases	none
Required?	false
Position?	3
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#)

### Examples

#### ----- EXAMPLE 1 -----

Description

-----

The first command gets all class instances in the Contoso.com domain and stores them in the \$Instances variable.

The second command gets the task with the display name "Get Monitor State" and starts the task for each class instance stored in the \$Instances variable.

```
PS C:\>$Instances = Get-SCOMClassInstance -Name *.Contoso.com
PS C:\>Get-SCOMTask -DisplayName "Get Monitor State" | Start-SCOMTask -Instance $Instances
```

#### ----- EXAMPLE 2 -----

Description

-----

The first command creates a hash table with a value for the Timeout parameter and stores it in the \$Overrides variable.

The second command gets all class instances with the name "Contoso Service" and stores them in the \$Instances variable.

The third command gets the task with the display name "Start NT Service" and starts the task for each class instance stored in the \$Instances variable.

The fourth command starts the task for each class instance stored in the \$Instances variable using a Timeout parameter of 60.

```
PS C:\>$Overrides = @{Timeout=60}
PS C:\>$Instances = Get-SCOMClassInstance -DisplayName "Contoso Service"
PS C:\>$Task = Get-SCOMTask -DisplayName "Start NT Service"
```

```
PS C:\>Start-SCOMTask -Task $Task -Instance $Instances -Override $Overrides
```

### ----- EXAMPLE 3 -----

Description

-----

The first command prompts the user for a user name and password, and stores the credential object in the \$Creds variable.

The second command gets a class instance named "Server01.Contoso.com" and stores it in the \$Instance variable.

The third command gets a task with the display name "Reset State" and starts the task for the class instance stored in the \$Instance variable. The command runs the task under the credentials stored in the \$Creds variable.

```
PS C:\>$Creds = Get-Credential
```

```
PS C:\>$Instance = Get-SCOMClassInstance -Name Server01.Contoso.com
```

```
PS C:\>Get-SCOMTask -DisplayName "Reset State" | Start-SCOMTask -Instance $Instance -  
TaskCredentials $Creds
```

## Related topics

[Get-SCOMClassInstance](#)

[Get-SCOMTask](#)

[Get-SCOMTaskResult](#)



# Test-SCOMCEIP

---

## Test-SCOMCEIP

Determines if the SCOM CEIP is enabled on the local computer.

### Syntax

Parameter Set: Empty

```
Test-SCOMCEIP [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession  
<Connection[]> ] [ <CommonParameters>]
```

### Detailed Description

The Test-SCOMCEIP cmdlet determines if the Operations Manager Customer Experience Improvement Program (CEIP) is enabled on the local computer.

If \$true is returned, CEIP is enabled. If \$false is returned, CEIP is disabled.

### Parameters

#### **-ComputerName<String[]>**

Specifies a computer to establish a connection with. The computer must be running the System Center Data Access service. The default is the computer for the current management group connection.

Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

#### **-Credential<PSCredential>**

Specifies a user account under which the management group connection will run. It must have access to the server that is specified in the ComputerName parameter if specified. The default is the current user.

Enter a PSCredential object, such as one that is returned by the Get-Credential cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection.

Enter a Management group connection object, such as one returned by the Get-SCManagementGroupConnection cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#)

## **Examples**

### **----- EXAMPLE 1 -----**

This command returns the result of whether CEIP is enabled on the local computer.

```
PS C:\>Test-SCOMCEIP
```

# Test-SCOMOperationalDataReporting

---

## Test-SCOMOperationalDataReporting

Determines whether operational data reporting is enabled for the management group.

### Syntax

Parameter Set: Default

```
Test-SCOMOperationalDataReporting [-ComputerName <String[]> ] [-Credential <PSCredential> ]  
[-SCSession <Connection[]> ] [ <CommonParameters>]
```

### Detailed Description

The Test-SCOMOperationalDataReporting cmdlet determines whether operational data reporting is enabled for the management group.

If \$true is returned, operational data reporting is enabled. If \$false is returned, operational data reporting is disabled.

### Parameters

#### **-ComputerName<String[]>**

Specifies a computer to establish a connection with. The computer must be running the System Center Data Access service. The default is the computer for the current management group connection.

Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-Credential<PSCredential>**

Specifies a user account under which the management group connection will run. It must have access to the server that is specified in the ComputerName parameter if specified. The default is the current user.

Enter a PSCredential object, such as one that is returned by the Get-Credential cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection.

Enter a Management group connection object, such as one returned by the Get-SCManagementGroupConnection cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#)

## **Examples**

### **----- EXAMPLE 1 -----**

This command returns the result of whether operational data reporting is enabled.

```
PS C:\>Test-SCOMOperationalDataReporting
```



# Uninstall-SCOMAgent

---

## Uninstall-SCOMAgent

Uninstalls agents from agent-managed computers.

### Syntax

Parameter Set: Empty

```
Uninstall-SCOMAgent -Agent <AgentManagedComputer[]> [-ActionAccount <PSCredential> ] [-PassThru] [ <CommonParameters>]
```

### Detailed Description

The Uninstall-SCOMAgent cmdlet uninstalls agents from agent-managed computers.

### Parameters

#### **-ActionAccount<PSCredential>**

Specifies the credentials under which the agent uninstall task will run. If this parameter is omitted or has a null value, the default action account of the management server managing the agent is used.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

#### **-Agent<AgentManagedComputer[]>**

Specifies one or more agent objects to uninstall. Enter a variable that represents the agents, or type a command that gets the agents.

For information about how to get agents, type `Get-Help Get-SCOMAgent`.

Aliases	none
Required?	true

Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-PassThru**

Returns an object representing the agent. By default, this cmdlet does not generate any output.

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](#)

## **Inputs**

The input type is the type of the objects that you can pipe to the cmdlet.

- **Microsoft.EnterpriseManagement.Administration.AgentManagedComputer**

## **Outputs**

The output type is the type of the objects that the cmdlet emits.

- **Microsoft.EnterpriseManagement.Administration.AgentTaskResult**

## **Examples**

### **----- EXAMPLE 1 -----**

Description

-----

This command gets the agent "Server01.Contoso.com" and then uninstalls it.

```
PS C:\>Get-SCOMAgent Server01.Contoso.com | Uninstall-SCOMAgent
```

## ----- EXAMPLE 2 -----

Description

-----

The first command gets the agent-managed computer named "Server01.Contoso.com" and stores it in the \$Agent variable.

The second command uninstalls the agent stored in the \$Agent variable.

```
PS C:\>$Agent = Get-SCOMAgent Server01.Contoso.com
```

```
PS C:\>Uninstall-SCOMAgent -Agent $Agent
```

## ----- EXAMPLE 3 -----

Description

-----

The first command gets the agent named "Server01.Contoso.com" and stores it in the \$Agent variable.

The second command uninstalls the agent stored in the \$Agent variable using the credentials the user is prompted for.

```
PS C:\>$Agent = Get-SCOMAgent Server01.Contoso.com
```

```
PS C:\>Uninstall-SCOMAgent -Agent $Agent -ActionAccount (Get-Credential)
```

## Related topics

[Get-SCOMAgent](#)

[Install-SCOMAgent](#)



# Update-SCOMADAgentAssignment

---

## Update-SCOMADAgentAssignment

Updates certain settings of the Active Directory agent assignment.

### Syntax

Parameter Set: FromAgentAssignment

```
Update-SCOMADAgentAssignment [[-LdapQuery] <String> ] -AgentAssignment <AgentAssignment> -  
PrimaryServer <ManagementServer> [-ComputerName <String[]> ] [-Credential <PSCredential> ]  
[-Exclude <String[]> ] [-PassThru] [-SCSession <Connection[]> ] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

### Detailed Description

The Update-SCOMADAgentAssignment cmdlet updates certain settings of the Active Directory agent assignment. Only the LDAP query and exclusion list can be modified. To make other changes, the agent assignment must be deleted and re-created.

### Parameters

#### **-AgentAssignment<AgentAssignment>**

The AD agent assignment.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

#### **-ComputerName<String[]>**

Specifies a computer to establish a connection with. The computer must be running the System Center Data Access service. The default is the computer for the current management group connection.

Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

**-Credential<PSCredential>**

Specifies a user account under which the management group connection will run. It must have access to the server that is specified in the ComputerName parameter if specified. The default is the current user.

Enter a PSCredential object, such as one that is returned by the Get-Credential cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

**-Exclude<String[]>**

The new list of computers to exclude, even if they are returned by the LDAP query. The existing exclusion list will be kept if this parameter is omitted.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-LdapQuery<String>**

The new LDAP query. The existing LDAP query will be kept if this parameter is omitted.

Aliases	none
Required?	false
Position?	3
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-PassThru**

If specified, the agent assignment will be returned to the pipeline.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-PrimaryServer<ManagementServer>**

The primary management server used for the agent assignment being updated. This parameter is automatically populated when you pipe input from Get-SCOMADAgentAssignment.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByPropertyName)
Accept Wildcard Characters?	false

## **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection.

Enter a Management group connection object, such as one returned by the Get-SCManagementGroupConnection cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-Confirm**

Prompts you for confirmation before executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#)

## Examples

### ----- EXAMPLE 1 -----

This command updates the LDAP query of an Active Directory agent assignment for domain contoso.com to return only computers with names matching "SQLSERVER\*".

```
PS C:\>Get-SCOMADAgentAssignment -Domain "contoso.com" | Update-SCOMADAgentAssignment -LdapQuery '(&!(sAMAccountType=805306369)(name=SQLSERVER*))'
```

# Update-SCOMLocation

---

## Update-SCOMLocation

Updates the settings of a geographic location instance.

### Syntax

Parameter Set: FromLocation

```
Update-SCOMLocation [-Location] <EnterpriseManagementObject> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-DisplayName <String> ] [-Latitude <String> ] [-Longitude <String> ] [-PassThru] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

### Detailed Description

The Update-SCOMLocation cmdlet updates the settings of a geographic location instance.

### Parameters

#### **-ComputerName<String[]>**

Specifies a computer to establish a connection with. The computer must be running the System Center Data Access service. The default is the computer for the current management group connection.

Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

#### **-Credential<PSCredential>**

Specifies a user account under which the management group connection will run. It must have access to the server that is specified in the ComputerName parameter if specified. The default is the current user.

Enter a PSCredential object, such as one that is returned by the Get-Credential cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-DisplayName<String>**

Specifies the display name of the location.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-Latitude<String>**

Specifies the latitude for a location in decimal degrees.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-Location<EnterpriseManagementObject>**

Specifies the location instance to update. To get a location instance, use the Get-SCOMLocation cmdlet.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-Longitude<String>**

Specifies the longitude for a location in decimal degrees.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-PassThru**

Indicates that the updated location instance is returned to the pipeline.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection.

Enter a Management group connection object, such as one returned by the Get-SCManagementGroupConnection cmdlet.



Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#)

### Examples

----- **EXAMPLE 1** -----

This command updates the latitude and longitude of the Los Angeles, CA location instance.

```
PS C:\>Get-SCOMLocation -DisplayName "Los Angeles, CA" | Update-SCOMLocation -Latitude 33.942809 -Longitude -118.4047064
```

# Update-SCOMRunAsAccount

---

## Update-SCOMRunAsAccount

Updates the credentials of an Operations Manager Run As account.

### Syntax

Parameter Set: ActionAccount

```
Update-SCOMRunAsAccount [-ActionAccount] <ActionAccountSecureData> [-RunAsCredential]
<PSCredential> [-AccountType <String> ] [-ComputerName <String[]> ] [-Credential
<PSCredential> ] [-PassThru] [-SCSession <Connection[]> ] [-Confirm] [-WhatIf] [
<CommonParameters>]
```

Parameter Set: BasicAccount

```
Update-SCOMRunAsAccount [-BasicAccount] <BasicCredentialSecureData> [-RunAsCredential]
<PSCredential> [-AccountType <String> ] [-ComputerName <String[]> ] [-Credential
<PSCredential> ] [-PassThru] [-SCSession <Connection[]> ] [-Confirm] [-WhatIf] [
<CommonParameters>]
```

Parameter Set: BinaryAccount

```
Update-SCOMRunAsAccount [-BinaryAccount] <GenericSecureData> [-Path] <String> [-AccountType
<String> ] [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-PassThru] [-SCSession
<Connection[]> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: CommunityStringAccount

```
Update-SCOMRunAsAccount [-CommunityStringAccount] <CommunityStringSecureData> [-
CommunityString] <SecureString> [-AccountType <String> ] [-ComputerName <String[]> ] [-
Credential <PSCredential> ] [-PassThru] [-SCSession <Connection[]> ] [-Confirm] [-WhatIf] [
<CommonParameters>]
```

Parameter Set: DigestAccount

```
Update-SCOMRunAsAccount [-DigestAccount] <SimpleCredentialSecureData> [-RunAsCredential]
<PSCredential> [-AccountType <String> ] [-ComputerName <String[]> ] [-Credential
<PSCredential> ] [-PassThru] [-SCSession <Connection[]> ] [-Confirm] [-WhatIf] [
<CommonParameters>]
```

Parameter Set: SimpleAccount

```
Update-SCOMRunAsAccount [-SimpleAccount] <SimpleCredentialSecureData> [-RunAsCredential]
<PSCredential> [-AccountType <String> ] [-ComputerName <String[]> ] [-Credential
<PSCredential> ] [-PassThru] [-SCSession <Connection[]> ] [-Confirm] [-WhatIf] [
<CommonParameters>]
```

Parameter Set: WindowsAccount

```
Update-SCOMRunAsAccount [-WindowsAccount] <WindowsCredentialSecureData> [-RunAsCredential]
<PSCredential> [-AccountType <String> ] [-ComputerName <String[]> ] [-Credential
<PSCredential> ] [-PassThru] [-SCSession <Connection[]> ] [-Confirm] [-WhatIf] [
<CommonParameters>]
```

## Detailed Description

The Update-SCOMRunAsAccount cmdlet updates the credentials of an Operations Manager Run As account. SCX cross-platform accounts and SNMPv3 accounts are not supported.

## Parameters

### **-AccountType<String>**

Specifies the type of the Run As account. This value is usually automatically populated by the pipeline.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByPropertyName)
Accept Wildcard Characters?	false

### **-ActionAccount<ActionAccountSecureData>**

Specifies an action account Run As account.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-BasicAccount<BasicCredentialSecureData>**

Specifies a basic Run As account.

Aliases	none
Required?	true
Position?	1
Default Value	none

Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-BinaryAccount<GenericSecureData>**

Specifies a binary Run As account.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-CommunityString<SecureString>**

Specifies the new community string for a community string account.

Aliases	none
Required?	true
Position?	2
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-CommunityStringAccount<CommunityStringSecureData>**

Specifies a community string Run As account.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	true (ByValue)

Accept Wildcard Characters?	false
-----------------------------	-------

### **-ComputerName<String[]>**

Specifies a computer to establish a connection with. The computer must be running the System Center Data Access service. The default is the computer for the current management group connection.

Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Credential<PSCredential>**

Specifies a user account. It must have access to the server that is specified in the ComputerName parameter if specified. The default is the current user.

Enter a PSCredential object, such as one that is returned by the Get-Credential cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-DigestAccount<SimpleCredentialSecureData>**

Specifies a digest Run As account.

Aliases	none
Required?	true

Position?	1
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-PassThru**

Returns the updated Run As account object. By default, this cmdlet does not generate any output.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Path<String>**

Specifies the path to a file that contains new credential data for binary accounts.

Aliases	none
Required?	true
Position?	2
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-RunAsCredential<PSCredential>**

Specifies new credentials for account types that use a user name and password.

Aliases	none
Required?	true
Position?	2

Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-SCSession<Connection[]>**

Specifies a connection to a management server. The default is the current management group connection.

Enter a Management group connection object, such as one returned by the Get-SCManagementGroupConnection cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-SimpleAccount<SimpleCredentialSecureData>**

Specifies a simple Run As account.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-WindowsAccount<WindowsCredentialSecureData>**

Specifies a Windows Run As account.

Aliases	none
Required?	true

Position?	1
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-Confirm**

Prompts you for confirmation before executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#)

## **Examples**

### **----- EXAMPLE 1 -----**

The first command gets the Run As account object named Domain Admin and stores the object in the \$WindowsAccount variable. The second command passes the Run As account stored in



\$WindowsAccount to Update-SCOMRunAsAccount which updates the credentials for the account to the user name and password provided by the user when prompted by Get-Credential.

```
PS C:\> $WindowsAccount = Get-SCOMRunAsAccount -Name "Domain Admin"
```

```
PS C:\>$WindowsAccount | Update-SCOMRunAsAccount -RunAsCredential (Get-Credential)
```

## ----- EXAMPLE 2 -----

The first two commands create a string containing an action account and a secure string containing a password. The third command creates a PSCredential object using the name and password from the first two commands and stores the object in the \$newCred variable. The last command gets the action account named SCOM Action Account and uses the pipeline operator to pass the account to Update-SCOMRunAsAccount which updates the account with the credentials stored in \$newCred.

```
PS C:\> $userName = "contoso\SCOMActionAccount"
```

```
PS C:\>$password = Read-Host -AsSecureString
```

```
PS C:\>$newCred = new-object System.Management.Automation.PSCredential $userName,$password
```

```
PS C:\>Get-SCOMRunAsAccount -Name "SCOM Action Account" | Update-SCOMRunAsAccount -RunAsCredential $newCred
```

## ----- EXAMPLE 3 -----

This command gets the community string account named MyCommunityStringAccount and uses the pipeline operator to pass the account to Update-SCOMRunAsAccount which updates the community string with the value supplied by the user when prompted by Read-Host.

```
PS C:\> Get-SCOMRunAsAccount -Name "MyCommunityStringAccount" | Update-SCOMRunAsAccount -CommunityString (Read-Host -AsSecureString)
```

## ----- EXAMPLE 4 -----

This example gets the binary account object named MyBinaryAccount and uses the pipeline operator to pass the object to Update-SCOMRunAsAccount which indicates that the new account data is located in the file named data.txt by using the Path parameter.

```
PS C:\> Get-SCOMRunAsAccount -Name "MyBinaryAccount" | Update-SCOMRunAsAccount -Path '.\data.txt'
```

# Write-SCOMCommand

---

## Write-SCOMCommand

Writes a new PowerShell command to interact with Operations Manager.

### Syntax

Parameter Set: Name

```
Write-SCOMCommand -Name <String> [ <CommonParameters>]
```

Parameter Set: NoName

```
Write-SCOMCommand -NoName [ <CommonParameters>]
```

### Detailed Description

The Write-SCOMCommand cmdlet writes a new PowerShell command to interact with Operations Manager. Uses the same common SDK connection code as the standard Operations Manager cmdlets.

### Parameters

#### **-Name<String>**

The name of the function to create.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

#### **-NoName**

If set, will create an unnamed script.

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 -----

This command outputs a new script function template named Test-ForMyManagementPack.

```
PS C:\>Write-SCOMCommand -Name 'Test-ForMyManagementPack'
```